

Esko Liimatainen

Ohjelmoitava maailma – SDx-tuotekehitysympäristö pilvipalvelin-, SDN/NFV- ja IoT-teknologioin

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

1.2.2017

Tekijä	Esko Liimatainen
Otsikko	SDx-tuotekehitysympäristö pilvipalvelin-, SDN/NFV- ja IoT-teknologioin
Sivumäärä	103 sivua
Aika	1.2.2017
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotuotanto
Ohjaaja	Yliopettaja Matti Puska
<p>Insinööriyön tavoitteena oli toteuttaa startup-projekti, jossa kehitetään ja otetaan käyttöön SDx-tuotekehitysalustan ensimmäinen versio. Alusta mahdollistaa sellaisten digitalisaatio 2.0 -palvelujen kehittämisen, jotka liittyvät muun muassa SmartCity-, SmartEnergy- ja SmartTransport-palveluihin. Alustan etuihin kuuluu, että se on avoin ja omassa hallinnassa sekä laitetuotteista riippumaton.</p> <p>Työssä tutkittiin digitalisaatio 2.0:n mahdollistavia SDx- eli pilvipalvelin-, SDN/NFV- ja IoT-teknologioita. Työssä tutkittiin teknologioiden kypsyyssastetta ja markkinanäkymää sekä valittiin tuotekehityspolku tuoteperheiden rakentamiseksi. Tutkimus tehtiin kirjallisuuskäytöksi ja Princetonin yliopiston SDN/NFV-verkkokurssimateriaaliin nojautuen. Valintaperusteina olivat pienet investointi- ja käyttökulut (CAPEX/OPEX), avoimien ohjelmistojen hyödyntäminen, maailmanlaajuisesti skaalautuva ja modulaarinen tuotteistaminen eri alustoille, kuten mobiililaitte-, PC-, palvelin- ja pilvialustoille sekä nopea tuotekehityssykli.</p> <p>Insinööriyössä päädyttiin kriteereiden pohjalta valitsemaan OpenStack-pilvipalvelimeen pohjautuva FIWARE-alusta ja sen ympärille muodostunut avoin ekosysteemi. FIWARE-alusta on Euroopan unionin osarahoittama ja yhteistyöyritysten ylläpitämä. FIWAREn SmartCity IoT -alustaa parannettiin laajentamalla sitä niin sanotulla Fog- eli sumuverkolla, joka ulottuu Espanjan Madridista Suomen pääkaupunkiseudun NEBULA OpenStack -pilveen asti. Sen jälkeen rakennettiin Espooseen SDx-laboratorioympäristö ja lisättiin SDN-kytkin ja NFV "middlebox" -tuotekehitysympäristöt sekä IoT-gateway- ja sensorilaitteistot. SDx-laboratorioympäristön SDN-kytkimeksi valittiin ONOS-verkonhallintaohjelmisto ja Open vSwitch -kytkin, koska se tukee hyvin operaattoreiden ajatusmallia ja mahdollistaa kehittyneimmän SDN/NFV-tietoliikenneverkkoarkkitehtuurin. NFV "middlebox" -tuotekehitysympäristöksi valittiin PC-ympäristössä toimiva Princetonin yliopiston SDN/NFV-verkkokurssiympäristö, jonka konfigurointi tehtiin Vagrant-ohjelmistolla. IoT-laitteistoina käytettiin Raspberry Pi- ja Arduino-kehityskittejä. Ohjelmistoina niissä käytettiin muun muassa Adafruit IoT -ohjelmistoja. Työssä päästiin tavoitteeseen: siinä tutkittiin, valittiin, suunniteltiin ja toteutettiin SDx-tuotekehitysalusta. Alustan jatkekehitys tehdään vuonna 2017 perustettavassa startup-yrityksessä.</p>	
Avainsanat	Digitalisaatio, tuotekehitys, SDx, pilvipalvelin, virtualisointi, SDN, NFV, IoT, Linux, EU's Digital Single Market (DSM), AIOTI, ETSI, FIWARE, SmartCities

Author	Esko Liimatainen
Title	Programmable world – the SDx R&D environment with cloud service, SDN/NFV and IoT technologies
Number of Pages	103 pages
Date	1 February 2017
Degree	Bachelor of Engineering
Degree Program	Information Technology
Specialisation option	Software Engineering
Instructors	Matti Puska, Principal Lecturer
<p>The aim of this study was to implement a start-up project in which the first version an SDx development platform was developed and deployed. The platform allows the development of digitalization 2.0 services-, relating to SmartCity, SmartEnergy and SmartTransport services. The advantages of the platform are that it is open and in users own control and also independent of device suppliers.</p> <p>Digitalization 2.0, which enables the SDx, such as cloud computing, SDN/NFV and IoT technologies were studied in the project as well. The degree of maturity and the market view of the technologies were studied, and the development path was chosen to build product families. The study was based on the literature sources and the Princeton University SDN/NFV online course material. The selection criteria were low investment and operating costs (CAPEX/OPEX), use of open source software, globally scalable and modular products for different platforms, such as mobile device, PC, server and cloud platforms, and rapid product development cycle.</p> <p>On the basis of criteria, it was decided to choose the FIWARE platform which is based on the OpenStack cloud server-, and an open ecosystem formed around FIWARE. The platform is co-funded by the European Union, and maintained by the partner companies. The FIWARE SmartCity IoT platform was improved by extending it by the Fog network which reaches from Madrid, Spain, to the NEBULA OpenStack cloud in the metropolitan area of Finland. After that SDx laboratory environment was built in Espoo, Finland, and SDN switch and NFV middlebox development environments and also IoT-gateway and sensor devices were added to SDx laboratory. The ONOS network management software with Open vSwitch was chosen for the SDN switch in the SDx laboratory environment-, because it supports well operators' way of thinking and allows the most advanced SDN/NFV telecommunication network architecture. The Princeton University SDN/NFV online course environment with configuration of the Vagrant software which works in PC environment was chosen for the development environment of the NFV middlebox. The Raspberry Pi and Arduino development kits were used as IoT devices. The Adafruit IoT software programs were used. In this study the targets were achieved. In the project, the SDx development platform was studied, selected, developed and implemented. Further development of the platform will take place in 2017 when a start-up company is established.</p>	
Keywords	Digitalization, R&D, SDx, cloud service, virtualization, SDN, NFV, IoT, Linux, EU's Digital Single Market (DSM), AIOTI, ETSI, FIWARE, SmartCities

Sisällys

Lyhenteet

1	Johdanto	1
2	Pilvipalvelinteknologia	2
2.1	Avoimet pilvipalvelinyhteisöt	2
2.1.1	Avoimien pilvipalvelinyhteisöjen vertailua	3
2.1.2	Avoim OpenStack-pilvipalvelinyhteisön ympäristö	8
2.2	Kaupalliset pilvipalvelinympäristöt	11
2.3	Virtualisoinnin eri toteutustavat	12
2.3.1	Täysvirtualisointi	12
2.3.2	Paravirtualisointi	13
2.3.3	Kevytvirtualisointi	13
2.3.4	Citrix XenServer -virtualisointiarkkitehtuuri	13
2.3.5	Windows Hyper-V -virtualisointiarkkitehtuuri	14
2.3.6	VMware ESXi -virtualisointiarkkitehtuuri	15
2.3.7	KVM-virtualisointiarkkitehtuuri ja Linux-jakelut	16
2.3.8	Kontaineri-virtualisointiarkkitehtuuri	17
3	SDN/NFV-teknologia	20
3.1	SDN – ohjelmisto-ohjatut verkot	20
3.2	SDN/NFV-teknologian yleistyminen	22
3.3	SDN/NFV:n kehityshistoria	23
3.4	SDN-sovelluskohteita	24
3.5	ONF – avoin verkkoyhteisö	25
3.6	OpenFlow-kontrollerit	26
3.7	OpenFlow-kommunikointiprotokolla	28
3.7.1	OpenFlow-vuotaulukko	29
3.7.2	OpenFlow 2.0 -protokolla - P4 ja POF Framework	32
3.8	OpenFlow-yhteyden virtualisointi	36
3.9	Mininet-verkkoemulaattoriympäristö	37
3.10	Avoimet SDN-kontrollerit	38
3.11	Avoimet SDN-kytkimet	42
3.11.1	White-Box-kytkinlaitteistoja	42

3.11.2	Open vSwitch -kytkinohjelmisto	43
3.11.3	Quagga-kytkinohjelmisto	43
3.12	Kaupalliset SDN-laitteistot	45
3.13	SDN/NFV-ohjelmoinnin haasteet	45
3.14	NV – verkon virtualisointi	46
3.15	NFV – verkkotoimintojen virtualisointi	48
3.16	OPNFV – avoin NFV-yhteisö	49
4	IoT-teknologia	51
4.1	IoT – esineiden internet	51
4.2	IoT:n yleistyminen	52
4.3	IoT-sovelluskohteita	54
4.4	IoT-alan toimijoita Suomessa	55
4.5	IoT-maisemakuva	55
4.6	IoT-ekosysteemi	56
4.7	IoT-protokollia ja standardeja	57
4.7.1	WLAN-standardi	58
4.7.2	Bluetooth-standardi	58
4.7.3	ZigBee-standardi	59
4.7.4	LoRaWAN-standardi	59
4.7.5	Wirepas-sensoriverkko	60
4.7.6	SIGFOX-sensoriverkko	60
4.7.7	4G- ja 5G-standardit	61
4.8	Teollinen internet 4.0	61
4.9	Euroopan yhtenäiset digitaaliset markkinat	64
4.9.1	Euroopan IoT-standardointi	64
4.9.2	FIWARE-alusta	65
4.9.3	FIWARE SmartCities	65
4.9.4	FIWAREn IoT-pino	66
5	SDx-tuotekehitysympäristön toteutus	68
5.1	SDx-ympäristön valinta	68
5.1.1	Pilvipalvelinympäristön valinta	68
5.1.2	SDN/NFV-ympäristön valinta	71
5.1.3	IoT-ympäristön valinta	72
5.1.4	SDx-ympäristön topologiakuvaus	72
5.2	SDx-ympäristön suunnittelu	73
5.2.1	Graafisen käyttöliittymän suunnittelu	73

5.2.2	Pilvipalvelinympäristön suunnittelu	73
5.2.3	SDN/NFV-ympäristön suunnittelu	74
5.2.4	IoT-ympäristön suunnittelu	75
5.2.5	SDx-ympäristön arkkitehtuuri	76
5.3	SDx-ympäristön toteutus	78
5.3.1	Graafisen käyttöliittymän toteutus	78
5.3.2	Pilvipalvelinympäristön toteutus	79
5.3.3	SDN/NFV-ympäristön toteutus	79
5.3.4	IoT-ympäristön toteutus	82
5.4	SDx-ympäristön testaus	83
5.4.1	Graafisen käyttöliittymän testaus	83
5.4.2	Pilvipalvelinympäristön testaus	83
5.4.3	SDN/NFV-ympäristön testaus	84
5.4.4	IoT-ympäristön testaus	85
6	Yhteenveto	86
	Lähteet	89

Lyhenteet

SDx	Software Defined Anything. Ohjelmoitava maailma. SDx sisältää digitalisaation mahdollistavat pilvipalvelin-, SDN/NFV- ja IoT-tekniologiat.
SDN	Software-defined Networking. Ohjelmisto-ohjatut verkot. Menetelmä, jossa tietoliikenneverkon ohjaus (Control plane) ja datan siirto (Data plane) erotetaan toisistaan ja joita yhdistää esimerkiksi OpenFlow-kommunikointiprotokolla.
NFV	Network Functions Virtualization. Verkkotoimintojen virtualisointi. Menetelmä, jossa tietoliikenneverkon NFV "middlebox" -toiminnot suoritetaan virtuaalisessa koneessa, joka voi sijaita pilvipalvelimessä.
OpenFlow	Kommunikointiprotokolla, joka yhdistää ohjausosan (Control plane) ja datan siirto-osan (Data plane) toisiinsa ja mahdollistaa näin verkkokytkeytymen kautta kulkevan datan ohjauksen.
ONF	Open Networking Foundation. Käyttäjälähtöinen organisaatio, joka on tarkoitettu edistämään SDN:n käyttöönottoa.
IaaS	Infrastructure as a Service. IaaS-mallissa kolmannet osapuolet ylläpitävät laitteistoja, ohjelmistoja, palvelimia, tietovarastoja ja muita infrastruktuuri-komponentteja.
OpenStack	Ilmainen avoimen lähdekoodin pilvipalveluohjelmistoalusta. Käytetään ensisijaisesti IaaS-ratkaisuna.
KVM	Kernel-based Virtual Machine on Linux-ytimen virtualisointialusta. Liitetty Linux-ytimen päähaaraan helmikuussa 2007.
FIWARE	Future Internet Core Platform. Tulevaisuuden internetohjelmistoalusta. Euroopan unionin osarahoittama FIWARE-alusta. Kehittäjinä muun muassa eurooppalaisia laitevalmistajia ja yliopistoja.
ETSI	European Telecommunications Standards Institute. Eurooppalainen telealan standardisointijärjestö.

IETF	Internet Engineering Task Force. Internetprotokollien standardisoinnista vastaava organisaatio.
IoT	Internet of Things. Esineiden internet. Suomessa käytetään myös käsitettä teollinen internet. Muodostuu sensoriverkosta, jossa välitetään sensoreilta saatava tapahtumatieto palvelimelle.
ZigBee	Sensoriverkkoteknologia, jolla tarkoitetaan IEEE 802.15.4 -standardin mukaista lyhyen kantaman tietoliikenneverkkoa (WPAN).
BLE	Bluetooth Low Energy. Bluetooth-versio 4.0, jossa on lisäominaisuutena käyttäjän tunnistus ja sijainti palvelimeen nähden.
WPAN	Wireless Personal Area Network. Langaton verkko, jolla voidaan yhdistää henkilön läheisyydessä olevat laitteet toisiinsa. Verkkotekniikoina käytetään muun muassa Bluetooth- ja ZigBee-verkkoja.
REST	Representational State Transfer. HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmointirajapintojen (API) toteuttamiseksi.
ACL	Access Control List. Tietoliikenneverkkojen reitittimissä ja kytkimissä käytetty lista säännöistä, joilla sallitaan vain tietyn liikenteen pääsy laitteeseen.
TTL	Time To Live. Kuvaa tietoliikenneverkossa lähetettävän datan elinaikaa.
CAPEX/ OPEX	Capital expenditures (CAPEX), Operating expenses (OPEX). Investointi- ja käyttökulut.

1 Johdanto

Uusien tuotteiden ja palvelujen innovointi ja suunnittelu on tullut yhä tärkeämpään osaan suomalaisessa insinööriyössä. Uudistuville markkinoille tarvitaan uusia tuoteperheitä ja uusimmat teknologiat. Digitalisaation mahdollistavia teknologioita on monia. Yksi niistä on Software Defined Anything (SDx) [1], joka on yhteisnimitys tässä insinööriyössä käsiteltäville pilvipalvelin-, Software-defined Networking (SDN)- [3; 11–16;], Network Function Virtualization (NFV)- [115] ja Internet Of Things (IoT) [118] -teknologioille.

Tämän insinööriyön tarkoituksena on tutkia, valita, suunnitella ja toteuttaa ohjelmoitavan maailman eli SDx-startup-yrityksen tuotekehitysympäristö pilvipalvelin-, SDN/NFV- ja IoT-teknologioin. Tavoitteena on selvittää teknologioiden kypsyyssastetta ja markkinanäkymää sekä löytää tuotekehityspolku tuoteperheiden rakentamiseksi. Tutkimus tehdään kirjallisuuslähteisiin ja SDN/NFV-verkkokurssimateriaaliin nojautuen [4]. Valintaperusteena on valita teknologiat siten, että ne ovat kahden vuoden kuluttua yritystasolla toteutettavissa ja markkinanäkymän alkupäässä.

Suunniteltu SDx-tuotekehitysympäristö on mahdollista ottaa käyttöön myöhemmässä yrityksen startup-vaiheessa ottamatta tarkemmin kantaa siihen, mitä itse suunniteltavat SDN/NFV- ja IoT-tuotteet tulevat olemaan. Ne voivat olla kuluttajille tai yritysten käyttöön suunniteltuja tuotteita ja palveluita.

Kriteereinä tuotekehitysympäristön valinnoissa ovat pienet investointi- ja käyttökulut (CAPEX/OPEX), avoimien ohjelmistojen hyödyntäminen, globaalisti skaalautuva ja modulaarinen tuotteistaminen mobiililaitte-, PC-, palvelin- ja pilvialustoille sekä nopea tuotekehityssykli.

Hyödyntämällä pilvipalvelin- ja SDN/NFV-teknologioita kiinteissä ja langattomissa tietoliikenneverkoissa voidaan pilvipalvelimesta käsin ylläpitää ja kehittää pienellä henkilöstömäärällä kustannustehokkaasti eri sovellusalueiden tuoteperheitä.

Pilvipalvelin-, SDN/NFV- ja IoT-teknologioiden sovellusalueita on runsaasti. Niitä ovat esimerkiksi kuluttaja-, IT-, Telecom- ja IoT-alueille suunnatut tuotteet ja palvelut. [7.]

2 Pilvipalvelinteknologia

Pilvipalvelimella tarkoitetaan muun muassa yksityisille ja yrityksille internetyhteyden välityksellä tarjottavaa palvelintietokoneen palvelua. Pilvipalvelujen palvelumallit ovat infrastruktuuri palveluna (Infrastructure As A Service, IaaS), alusta palveluna (Platform As A Service, PaaS) ja ohjelmisto palveluna (Software As A Service, SaaS) [8].

Tunnetun pilvipalvelinteknologian suunnannäyttäjän VMwaren tutkimuksen [110] mukaan IT-palvelut jakaantuvat maailmanlaajuisesti pilvipalvelimien ja perinteisten IT-palvelimien kesken taulukon 1 mukaisesti.

Taulukko 1. Pilvipalvelimen ja perinteisten IT-palveluiden käyttösuhteet vuosina 2006, 2016, 2021 ja 2030 VMwaren vuonna 2016 tekemän tutkimuksen mukaan [110].

IT-palvelut	2006	2016	2021	2030
<i>Pilvipalvelimet (%), joista</i>	2	27	50	81
- yksityiset pilvet (%)	0	12	20	29
- julkiset pilvet (%)	2	15	30	52
<i>Perinteiset IT-palvelimet (%)</i>	98	73	50	19

VMwaren tutkimuksen mukaan suurin osa IT-palvelujen tuottajista on seuraavan kymmenen vuoden kuluttua siirtynyt pilvipalvelimien käyttäjiksi.

2.1 Avoimet pilvipalvelinyhteisöt

Ylimmältä tasolta asioita tarkasteltaessa vastaan tulee erilaisia enemmän ja vähemmän avoimia pilvipalvelinyhteisöjä, joiden alla yliopistot, tiedeyhteisöt ja yritykset toimivat.

Tuotekehityksen pilvipalvelinympäristöä valittaessa on tärkeää ymmärtää, minkälaisia vaikuttajia, voimia ja teknologia-arkkitehtuureja eri pilvipalvelinympäristöjen takana on, ja tehdä teknologiaavalinta tältä pohjalta.

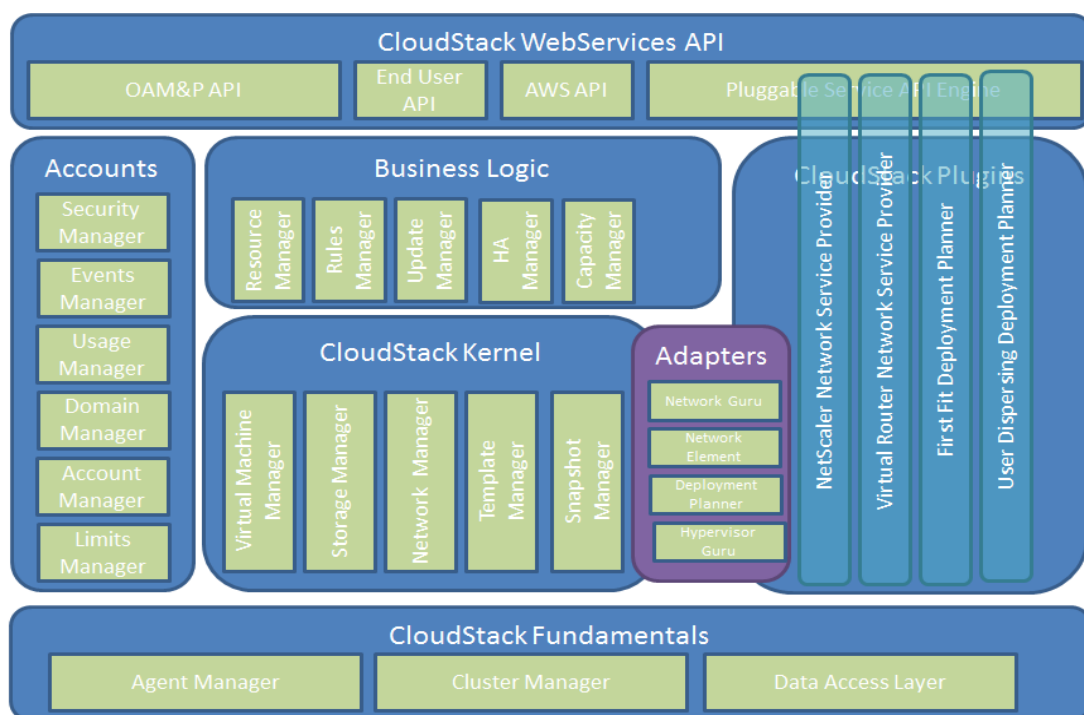
2.1.1 Avoimien pilvipalvelinyhteisöjen vertailua

Seuraavaksi tutkitaan suurimpia keskenään kilpailevia pilvipalveluita kehittäviä yhteisöjä, joita ovat muun muassa OpenStack, CloudStack, OpenNebula, Eucalyptus ja OpenCompute [17; 26; 29; 31; 32].

Ensimmäinen tarkasteltava pilvipalvelinyhteisö on **OpenStack**, jonka kehittämisessä ovat mukana sellaiset yritykset, kuin Rackspace (perustajana yhdessä NASAn kanssa vuonna 2010), IBM, HP, AT&T, Oracle, RedHat, Nebula, SUSE, Ubuntu ja Citrix [11]. Citrix on lopettanut toimintansa OpenStackissä vuonna 2012 ja jatkaa perustamassaan kilpailevassa CloudStackissä. OpenStackin kehittäjinä ovat lähinnä suuret IT-yritykset, yliopistot ja tiedeyhteisöt. OpenStack-arkkitehtuuri kuvataan tarkemmin luvussa 2.1.2.

Toinen avoimia pilvipalveluita kehittävä yhteisö on Citrixin **CloudStack**. CloudStack-yhteisöä kehittää Citrix. Se osti Cloud.com:n vuonna 2011 [26; 27]. Käyttäjinä ovat muun muassa Autodesk, China Telecom, Dell, Walt Disney, Huawei, Orange ja Tata [26].

Apache CloudStack (kuvassa 1) tukee useita hypervisoreita, kuten KVM, VMware vSphere ESXi, (aiemmin myös Hyper-V:tä) ja Citrix XenServer/XCP. Lisäksi se tukee Amazon Web Service (AWS) -ohjelmointirajapintaa (Application Programming Interface, API) ja OpenCloud-rajapintaa OpenGridForum-yhteisön kautta. CloudStackin koodi on julkaistu avoimen GPLv3:n ja Apache License 2.0:n alla. Ensimmäinen stabiili julkaistu versio on CloudStack 4.0.2 [27.]



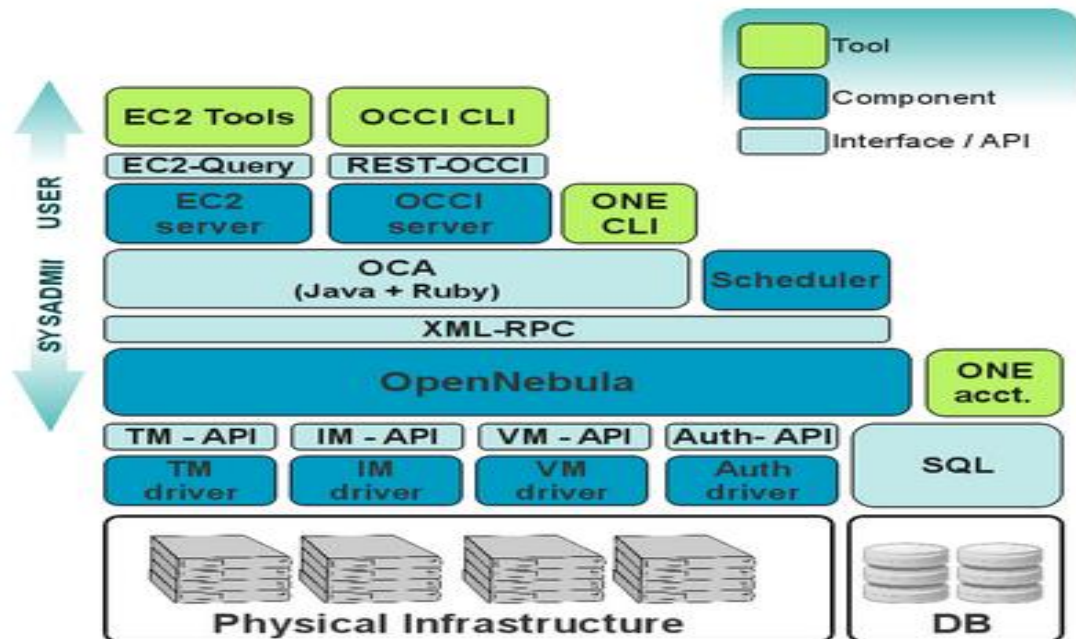
Kuva 1. CloudStack-ohjelmistoarkkitehtuuri [28].

CloudStack on kehittynyt kohti "plugin"-ohjelmistoarkkitehtuuria, jossa kolmannen osapuolen toimittajat voivat liittyä Java-pohjaisen "Pluggable Service API Engine" -rajapinnan kautta CloudStackiin.

Kolmas avoimia pilvipalveluita kehittävä yhteisö on **OpenNebula** [29]. Yhteisö on vaihtoehto OpenStackille, joka yhteisön mukaansa on fragmentoitunut, kehittymätön ja liian monimutkainen. VMware on puolestaan liian kallis ja joustamaton. OpenNebulan koodi on julkaistu Apache License 2.0:n alla.

Alkuperäisenä kehittäjäyhteyksinä oli C12G, joka vaihtoi nimensä OpenNebula Systemiksi. Nimi kuvaa paremmin yhtiön kansainvälistä päätuotetta OpenNebulaa. Yhtiö on lähtöisin Espanjan Madridista. Käyttäjinä ovat muun muassa Telefonica, Akamai, Blackberry ja ChinaMobile [28.]

OpenNebula 4.0 (kuvassa 2) tukee seuraavia hypervisoreita: KVM, Xen ja VMware ESXi. Se on ohjelmoitu muun muassa Java-, Ruby-, C- ja C++-kielillä.

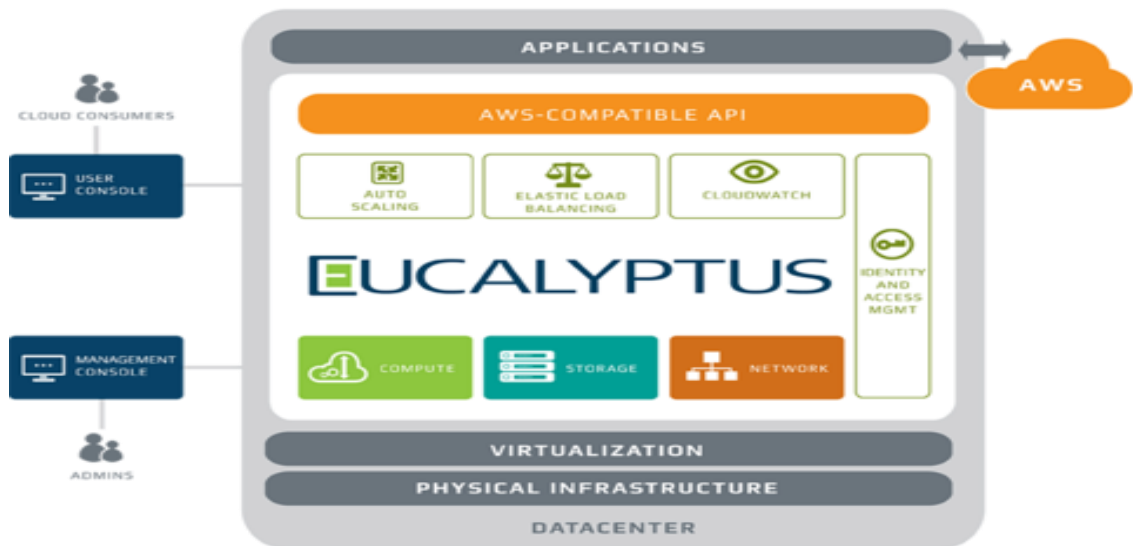


Kuva 2. OpenNebula-verkkoarkkitehtuuri [30].

Uusin versio OpenNebulasta on 4.4 Retina, jonka mukana tulee uusia ominaisuuksia, kuten tietokannan optimointia, monitorointia, hybrid-pilviratkaisu ja yleisiä pilvirajapintoja.

Neljäs avoimia pilvipalveluita kehittävä yhteisö on **Eucalyptus**. Yhteisö kehittää Amazon Web Services (AWS) yhteensopivaa "private"- ja "hybrid"-pilvipalvelu-ympäristöä. Eucalyptus tulee sanoista "Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems". Eucalyptus Systems Inc. julkaisi maaliskuussa 2012 AWS:n kanssa sopimuksen, jossa se lupautui antamaan tukensa AWS-yhteensopivuudelle. Eucalyptuksen on ostanut Hewlett-Packard. Kehittäjinä ovat useat yritykset, kuten Mosaik, AppDynamics, NASA, Nokia ja memsql. [31.]

Eucalyptus (kuvassa 3) tukee muun muassa seuraavia hypervisoreita: KVM, Xen ja VMware ESXi. Lisensoitu GPLv3 mahdollistaa ohjelmistojen uudelleen hyödynnettävyyden. Käyttöjärjestelmänä (Master host) on GNU/Linux, jonka alla voi toimia Linux- ja Windows-virtuaalikoneita. Ne on ohjelmoitu Java- ja C-kielillä. [31].



Kuva 3. Eucalyptus-verkkoarkkitehtuuri [31].

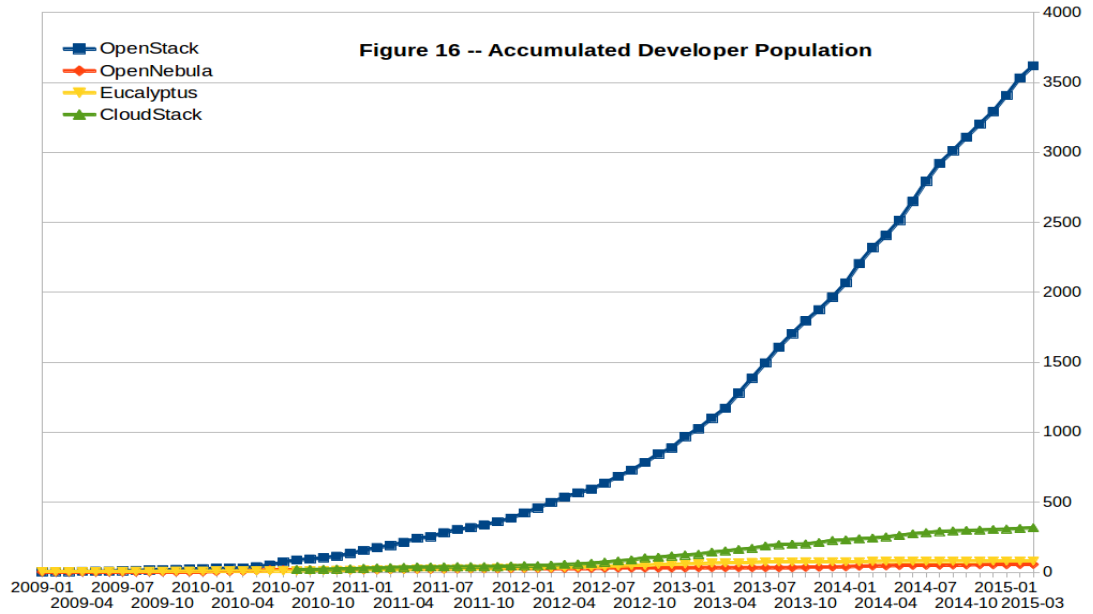
Hewlett-Packardin oston myötä tuotenimenä on ”HPE Helion Eucalyptus, Open source hybrid cloud software for AWS users”.

Viides avoimia pilvipalveluita ja myös laitteisoja kehittävä yhteisö on **OpenCompute** [32]. Yhteisö on kehittänyt muun muassa Facebookin käyttämän ”6-pack”-kytkimen, jossa on kaikki avointa, sekä ohjelmistot että laitteistot [33; 34]. Kehittäjinä ovat useat yritykset, kuten Facebook, Microsoft, Intel, Rackspace Hosting, Arista networks ja Goldman Sachs [32].

Yleisesti huomioitava asia on, että pilvipalveluja tarjoavat yritykset ovat rakentaneet erilaisia rajapintoja kilpailevien yritysten tuotteisiin, vaikka ne ovat eri kehittäjäyhteisöjen alla. Esimerkiksi VMware on rakentanut rajapinnan OpenStackin virtuaalikoneille, joten niitä voidaan suorittaa VMware hypervisor -pilvi-infrastuktuurin päällä.

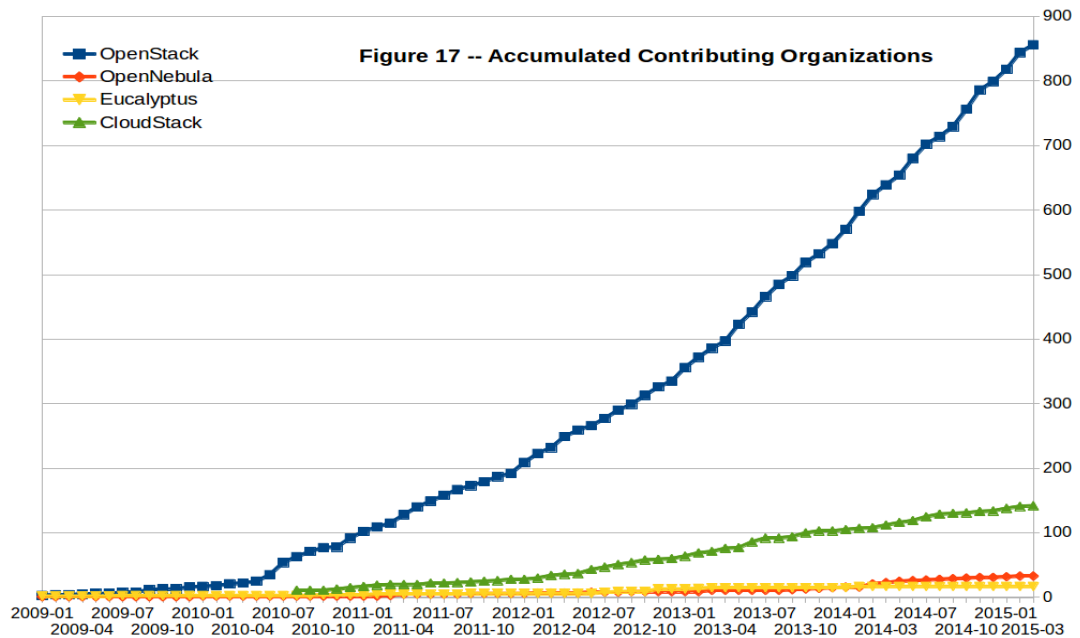
Australialaisen Sydneyn yliopiston (kuvat 4 ja 5) tutkimus, jossa vertaillaan kehittäjien ja organisaatioiden lukumäärää edellä esitellyissä pilvipalvelinyhteisöissä. Ylivoimaisesti suosituin kehittäjien ja organisaatioiden keskuudessa on OpenStack [35].

Kuvasta 4 nähdään, että OpenStackissä on eniten kehittäjiä [35].



Kuva 4. Kehittäjien lukumäärä eri pilviyhteisöissä [35].

Kuvasta 5 nähdään, että OpenStackissä on eniten sitä kehittäviä organisaatioita [35].



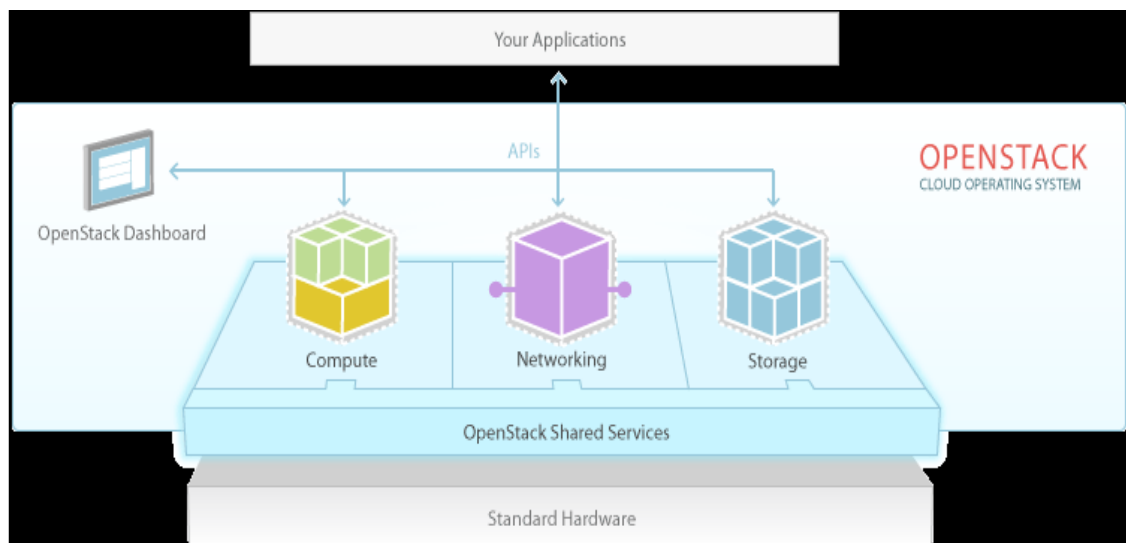
Kuva 5. Organisaatioiden lukumäärä eri pilviyhteisöissä [35].

2.1.2 Avoin OpenStack-pilvipalvelinyhteisön ympäristö

Avoimen OpenStack API -rajapinnan (kuva 6) alla oleva tuotekehityksen pilvipalvelinympäristö tulee suunnitella siten, että se yrityksen startup-vaiheen jälkeenkin palvelee hyvin ja skaalautuu mahdollisen liiketoiminnan kasvun myötä. [18.]

OpenStack-yhteisö tarjoaa kehyksen, jonka puitteissa yritykset voivat suunnitella pilvipalvelintuotteensa. OpenStack-yhteisön yritykset kilpailevat avoimella teknologiallaan IaaS-yritysasiakkaista VMwarea ja Citrixia vastaan. Amazon on myös jälleen tulossa vahvasti taistelemaan yritysasiakkaista lisäämällä tietoturvaominaisuuksia kaupallisiin tuotteisiinsa. OpenStack-kehityksessä mukana olo antaa yrityksille mahdollisuuden vaikuttaa avoimen API-rajapinnan suunnittelussa. [25.]

OpenStack on ilmainen ja avoimen lähdekoodin pilvipalvelualusta, jota käytetään ensisijaisesti infrastruktuuripalveluna (IaaS). OpenStackin API tarjoaa rajapinnan sen alla olevaan pilvipalveluun. OpenStackin käyttäjiä ovat muun muassa PayPal-verkkomaksu- ja Bloomberg-finaanssialan sekä telealan yhtiöt. [18.]



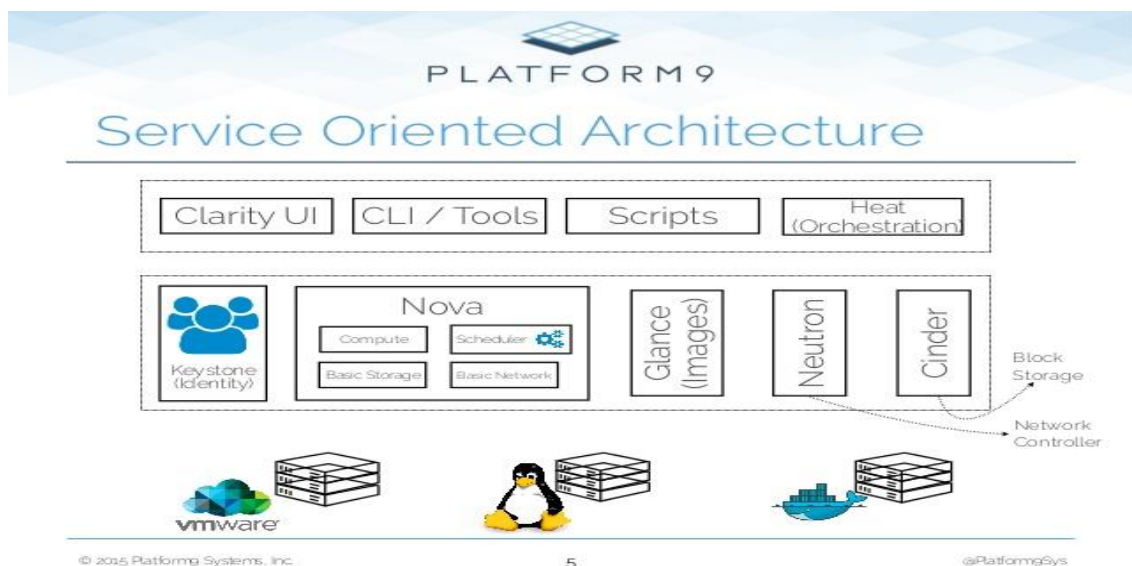
Kuva 6. OpenStack-verkkoarkkitehtuuri [18].

OpenStackin ilmaisversio-ohjelmistot vaativat IT-henkilöstöltä paljon aikaa ja osaamista muun muassa ohjelmistojen päivitysten osalta. Tältä voi välttyä käyttämällä kaupallisia Open-Stack-ohjelmistoja.

OpenStack on historiallisesti tukenut KVM-hypervisoria, joka on integroituna Linux-ytimeen. Erikseen komennolla asennettava QEMU-ohjelmisto tarjoaa sille muun muassa hallintarajapinnan. QEMU/KVM-tuki käyttää Linux-ytimen libvirtinrajapintaa. OpenStack tukee myös muita hypervisoreita. OpenStack Hypervisor -taulukkoon on koottu tuetut ominaisuudet [19].

OpenStack-arkkitehtuurin hallintaan on saatavilla kaupallisia graafisia hallintaohjelmistoja. Seuraavaksi kuvataan kolmea OpenStackiin pohjautuvaa esimerkkipilveä, jotka voisivat soveltua startup-yrityksen pilvipalvelun tarjoajiksi.

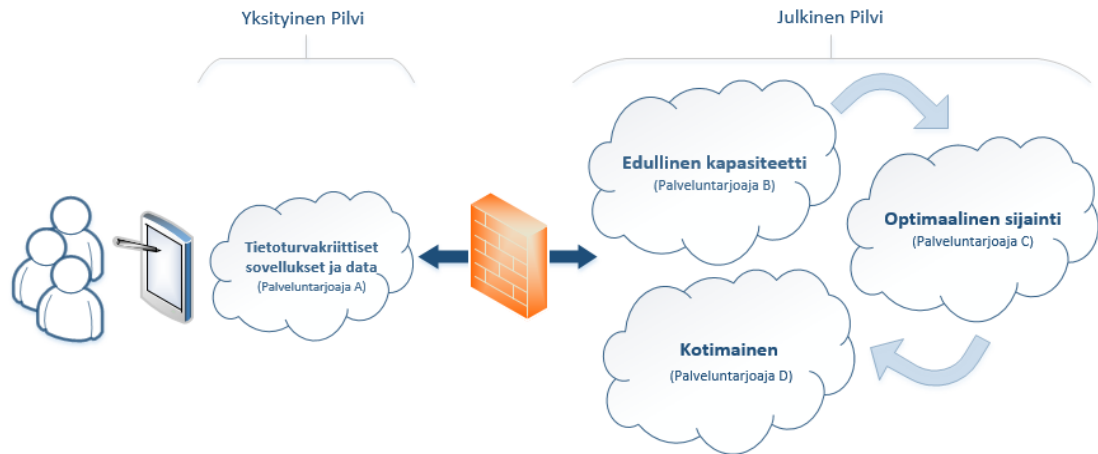
Ensimmäisenä esimerkkinä on Kaliforniasta lähtöisin oleva Platform9 OpenStack-manageri (kuva 7). Se on kehitetty lähinnä yrityksille, jotka haluavat tehdä jo olemassa olevista palvelimistaan yksityisen pilven, jota on helppo hallita. Se on tarkoitettu yrityksille, joilla on tai tulee olemaan noin 50–5000 palvelinta. Yhden henkilön käyttöön tarkoitettu tutustumisversio Lite ja varsinaiseen yrityskäyttöön tarkoitettu versio Enterprise on mahdollista ottaa tutustumiskäyttöön 15 päiväksi, jonka jälkeen on mahdollista tehdä vuosisopimus niiden kanssa. Hintatiedot ovat saatavissa ottamalla yhteyttä yritykseen. [20; 21.]



Kuva 7. Platform9 OpenStack-managerin arkkitehtuuri [22].

Platform9 OpenStack-managerin etuihin kuuluu, että sillä voi hallita useita virtualisointiympäristöjä, kuten VMware ESXi, KVM/Linux ja Docker.

Toisena esimerkkinä on suomalainen OpenStack-arkkitehtuuria hyödyntävä yritys Nebula, jonka pilvipalvelimet sijaitsevat Helsingin Pitäjänmäellä ja Lauttasaassa. Nebula tarjoaa kuvan 8 mukaisen hybridipilvipalvelinympäristön.



Kuva 8. Nebulan hybridipilvi [24].

Palveluntarjoaja A, joka voi olla yritys itse, huolehtii tietoturvallisesta materiaalista. B:n pilvi on edullisin julkinen pilvi suuriin tilapäisiin kapasiteettitarpeisiin. Palveluntarjoaja C sijaitsee ulkomailla lähellä käyttäjiä, ja D on kotimainen julkisen pilven palveluntarjoaja, jossa sijaitsee yrityksen muu data, esimerkiksi sähköposti ja verkkosivut. [24.]

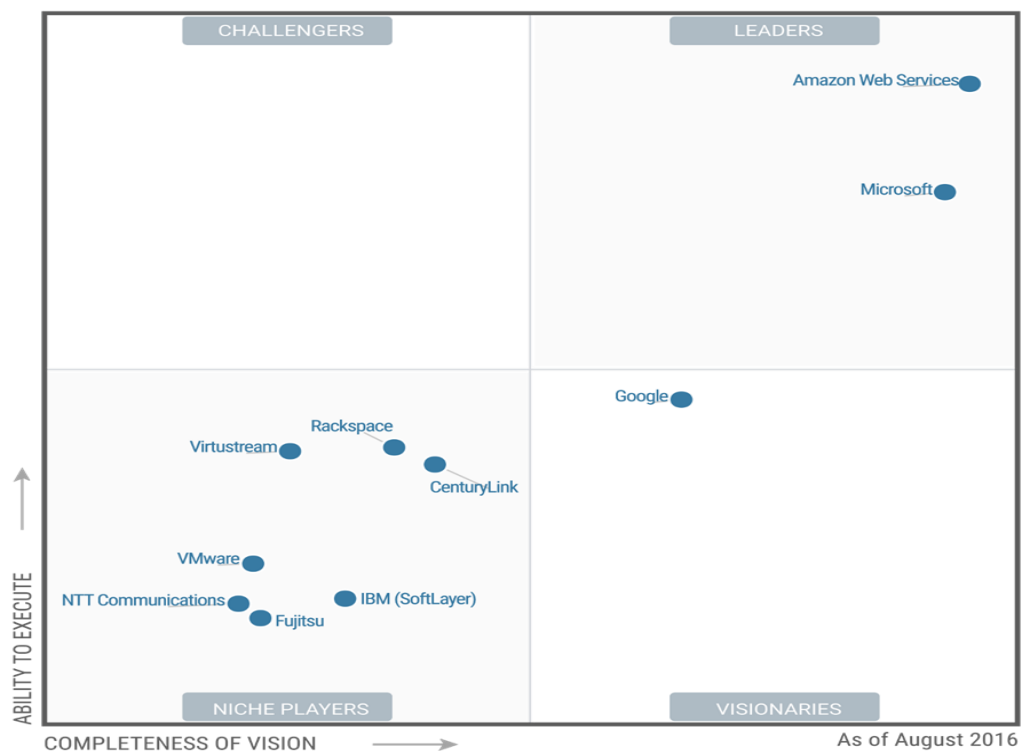
Kolmantena esimerkkinä on FIWARE-alusta. Euroopan unionin seitsemäs kehysohjelma (EU's Seventh Framework Programme, FP7) on luotu parantamaan Euroopan teollisuuden kilpailukykyä ja varmistamaan Euroopan globaali johtajuus ICT-teollisuudessa [155]. Tämä kehysohjelma osarahoittaa FIWARE-yhteisöä. Se on itsenäinen avoin yhteisö, jonka jäsenet ovat sitoutuneet rakentamaan julkisen ja maksuttoman sekä kestävän ekosysteemin. Tämä ekosysteemi tarjoaa muun muassa pienille ja keskisuurille yrityksille hyvän toimintaympäristön ja mahdollisuuden laajentua Euroopan ulkopuolelle FIWARE-yhteisön mukana. [154.]

FIWARE-alusta sisältää muun muassa OpenStack-pilvipalvelinympäristön, QE-MU/KVM-virtualisointiympäristön ja OpenFlow Network Information and Control (OF-NIC) -kontrollerin, joka on laajennus OpenDaylight-kontrolleriin. OpenDaylight-alusta sisältää myös Open Network Operating System (ONOS) -verkonhallintajärjestelmän, jota tutkitaan tarkemmin luvussa 4.9.2 FIWARE-alusta.

2.2 Kaupalliset pilvipalvelinympäristöt

Suurimpia kaupallisia pilvipalveluiden tarjoajia ovat Amazon Web Services (AWS) alustoina Amazon EC2 ja S3, Microsoft Azure alustoina Windows Server 2012 ja Hyper-V, Google Cloud Platform, VMware vSphere, Salesforce, IBM SmartCloud, HP Cloud, Oracle ja SAP [25].

Hyvän kokonaiskuvan kaupallisten suljettujen ja avoimien IaaS-pilvipalvelujen tarjoajien sijoittumisesta antaa Gartnerin ”Magic Quadrant” -analyysi [57], joka on vuodelta 2016 (kuva 9).



Kuva 9. Gartnerin ”Magic Quadrant”-analyysi IaaS-pilvipalvelutarjoajista vuodelta 2016 [57].

Suljettuja omia ohjelmistoja käyttävät muun muassa Amazon, Microsoft, Google ja VMware [57].

Avointa OpenStackiä käyttävät muun muassa Rackspace, IBM ja CenturyLink. Muut avoimet pilvipalvelinohjelmistot, esimerkiksi CloudStack ja OpenNebula, ovat tässä analyysissä marginaalisia. Eucalyptus kehittää Amazon Web Services (AWS) yhteen-

sopivaa avointa pilvipalvelinohjelmistoa. Virtualstream käyttää OpenStackia ja VMwarea pilvialustanaan. [57.]

Olennaisin muutos vuoden 2015 analyysiin [56] verrattuna on VMwaren sijoituksen huomattava heikentyminen. Samoin OpenStack-alustaa käyttävien IBM:n, CenturyLinkin, Rackspacen ja Virtualstreamin sijoitus on heikentynyt. Toisaalta Amazonin, Microsoftin ja Googlen johtoasemat ovat vahvistuneet.

Näistä IaaS-pilvipalvelintarjoajista monet, kuten VMware ja Microsoft, ovat laajentaneet tuotteisiinsa myös virtuaaliset SDN-kytkimet. Tästä kerrotaan luvuissa 3.12 Kaupalliset SDN-laitteistot ja 3.14 NV – verkon virtualisointi.

2.3 Virtualisoinnin eri toteutustavat

Tämä insinöörityö perustuu suurelta osin virtualisointitekniikoihin, ne ovat siten yksi tärkeimmistä tämän työn osakokonaisuuksista.

Yleisesti ottaen virtualisoinnilla tarkoitetaan tietotekniikassa jonkin fyysisen resurssin (prosessori, muisti, tallennuskapasiteetti, tietoliikenneverkko tai sen osa) jakamista useaan loogiseen resurssiin tai usean fyysisen resurssin yhdistämiseksi yhdeksi loogiseksi resurssiksi [36].

Fyysisiä laitteistoja, esimerkiksi yksityistä pilvipalvelinta (private cloud), rakennettaessa, on valittava oikeanlainen laitteistoarkkitehtuuri, esimerkiksi Intel VM- ja AMD-V-tekniologioiden välillä, sekä arkkitehtuurin päällä toimiva virtualisointialusta eli hypervisor.

2.3.1 Täysvirtualisointi

Täysvirtualisoinnissa eli 1-tyypin Bare-Metal-hypervisoreissa ohjelmistoja suoritetaan suoraan fyysisen laitteiston päällä, jolloin niiden välinen viestintä on nopeampaa ja turvallisempaa verrattuna paravirtualisointiin. Täysvirtualisointi on laitteistotason virtualisointia. [41.]

Täysvirtualisointiympäristöjä tarjoaa useampikin yritys. Tunnetuimpia ovat VMware ESXi, Microsoft Hyper-V ja Citrixin XenServer. Esimerkiksi Oracle VM Server käyttää OpenSource Xen hypervisor -tekniikkaa tuotteissaan. Suurin markkinaosuus on VMwarella niin ilmaisten kuin kaupallistenkin versioiden osalta. Edellä mainituista virtualisointiympäristöistä kaikki tukevat SDN:n OpenFlow'ta.

2.3.2 Paravirtualisointi

Paravirtualisoinnissa eli 2-tyypin Hosted-hypervisoreissa ohjelmistoja suoritetaan perinteisen esimerkiksi Linux- ja Windows-käyttöjärjestelmien päällä. Paravirtualisointi on käyttöjärjestelmätason virtualisointia. Sen etuna on käyttöönoton helppous, mutta heikkona puolena on hitaus ja alttius isäntäkäyttöjärjestelmän mahdollisille vikaantumisille [4; 41.] Tunnetuimpia 2-tyypin hypervisoreita ovat Oracle VM VirtualBox [98] ja uusimpana vuonna 2007 Linux-ytimeen integroitu KVM, joka on avoimen lähdekoodin ohjelmisto. Linux-käyttöjärjestelmä toimii siten isäntäkoneena, jonka päällä on KVM-hypervisor [42].

2.3.3 Kevytvirtualisointi

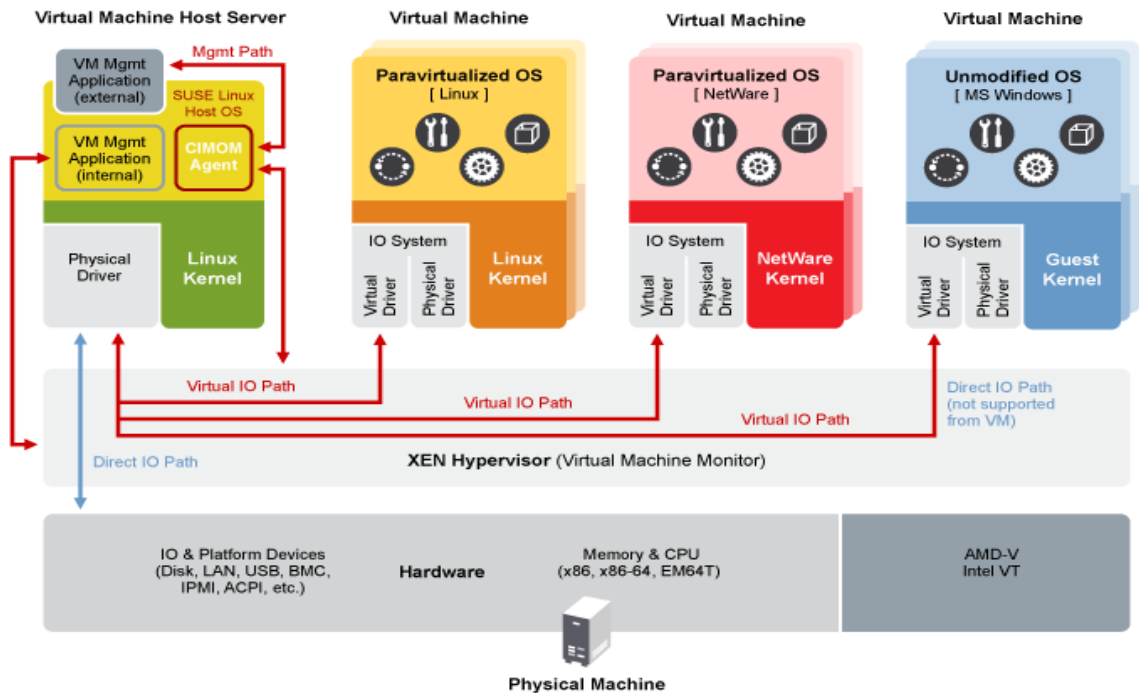
Linux Containerilla (LXC) virtualisoidaan Linux-ytimen tasoa. Kontainerin toiminta perustuu Linux-ytimen control groupiin (cgroups). Control group on kokoelma prosesseja, joita sitovat samat kriteerit. Nämä ryhmät voivat olla hierarkkisia, ja niissä jokainen ryhmä perii rajoitukset pääryhmältä. Linux-ydin tarjoaa pääsyn cgroup-rajapinnan kautta useille kontrollereille (alijärjestelmille): esimerkiksi "muisti"controller rajoittaa muistin käyttöä ja "cpuacct" jakaa prosessorin käyttöä [52.] Kontainereiden arkkitehtuuria ja toimintaa käsitellään perusteellisemmin luvussa 2.3.8 Kontaineri-virtualisointiarkkitehtuuri.

2.3.4 Citrix XenServer -virtualisointiarkkitehtuuri

Citrixin XenServer on avoimen lähdekoodin 1-tyypin hypervisor, jolla voi virtualisoida laitteistoa laajemmaltikin.

Citrix XenServer hyödyntää avoimen lähdekoodin projektia Xen Hypervisoria. XenServer on suunniteltu Windowsin ja Linuxin virtuaalikoneiden (VM) hallintaan (kuva 10).

XenServeriä kutsutaan ”thin layer” 1 -tyypin virtualisointialustaksi. Host Server eli ”Domain0” voi olla Linux, Solaris, BSD tai jokin muu palvelin.

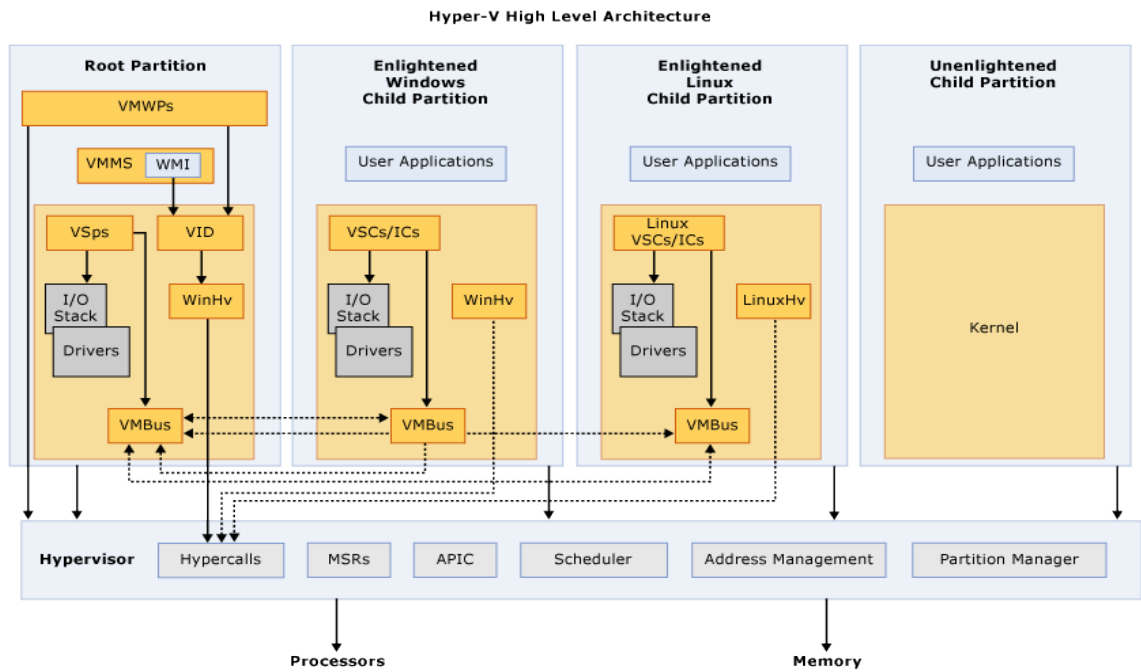


Kuva 10. XenServer-arkkitehtuuri SUSEn toteutuksella [39].

Kuvan 10 esimerkitapauksessa vasemmanpuoleinen Domain0 on Linux SUSE VM. Kaksi seuraavaa on paravirtualisoituja, DomainU (User) VM, eli ne toimivat Domain0:n päällä. Oikeanpuoleinen on täysvirtualisoitu VM, eli se toimii suoraan Xen Hypervisorin kautta laitteistoon.

2.3.5 Windows Hyper-V -virtualisointiarkkitehtuuri

Windows Hyper-V:n arkkitehtuuri (kuva 11) perustuu edellä esitettyyn Citrixin Xenin arkkitehtuuriin. Historiallisesti tarkasteltuna kyseisillä yrityksillä on ollut aiemmin syvälistä yhteistyötä, joka selittää tämän.

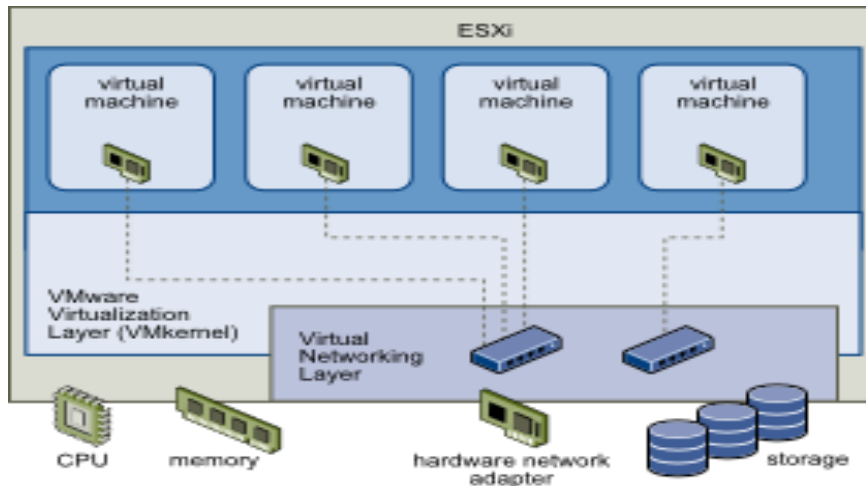


Kuva 11. Windows Hyper-V:n arkkitehtuuri [40].

Yhtäläisyydet XenServer-virtualisointiin ovat huomattavat. Domain0:na toimii Windows Server, ja DomainU:na voi olla Windows tai Linux VM [40].

2.3.6 VMware ESXi -virtualisointiarkkitehtuuri

VMware ESXi on "thick layer" -ohjelmisto ja 1-tyyppin hypervisor (kuva 12). Siitä käytetään myös nimityksiä BareMetal, embedded ja Native hypervisor. Tämä tarkoittaa sitä, että hypervisor toimii suoraan laitteiston päällä isäntäkoneena ja sen päällä ovat VM:t. [41.]



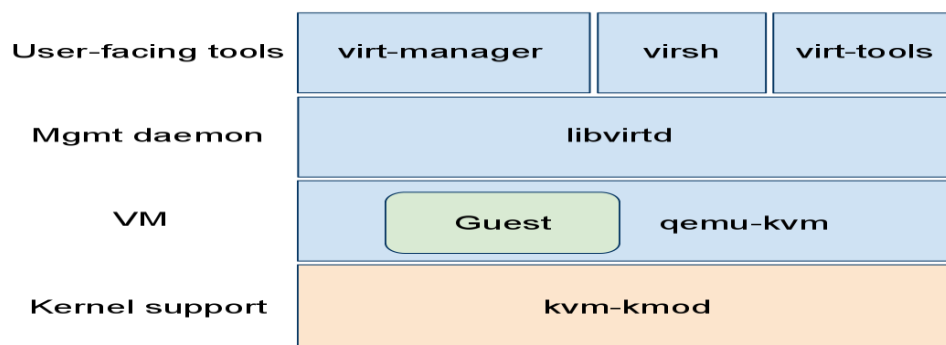
Kuva 12. VMware ESXi -arkkitehtuuri [41].

VMware ESXi:ssä ei ole Domain0-palvelinta, kuten esimerkiksi XenServerissä ja Hyper-V:ssä. VMware ESXi on hyvin laajalti käytetty virtualisointialusta palvelimissa ja PC:ssä. [41.]

2.3.7 KVM-virtualisointiarkkitehtuuri ja Linux-jakelut

KVM on integroituna Linux-ytimeen versiosta 2.6.20 alkaen ja toimii siellä kuten 1-tyyppin bare metal -hypervisor (kuva 13) [42].

Linux-käyttöjärjestelmässä QEMU (Quick Emulator) asennetaan komennolla `qemu-kvm <parametrit>`. QEMU ja KVM yhdessä mahdollistavat virtualisoinnin.



Kuva 13. KVM-arkkitehtuuri [43].

Esimerkiksi RedHatin pilvi- ja virtualisointiportfolio perustuu KVM:n virtualisointialustaan. Virtualisointialusta on nimetty RHEV-hypervisoriksi.

Yritysten käyttämistä Linux-jakeluista yleisin on RedHat Enterprise Linux (RHEL), jonka markkinaosuuden on arvioitu olevan 65–80 %:n luokkaa. RedHat tarjoaa stabiilin alustan, ja sen ansaintalogiikka perustuu tuen antamiseen yrityksille. RedHat-ilmaisjakeluja ovat CentOS ja Fedora [44]. CentOS on stabiilimpi ja palvelimissa käytetty. Fedoraa käytetään PC:ssä, ja se sisältää uusimpia ominaisuuksia.

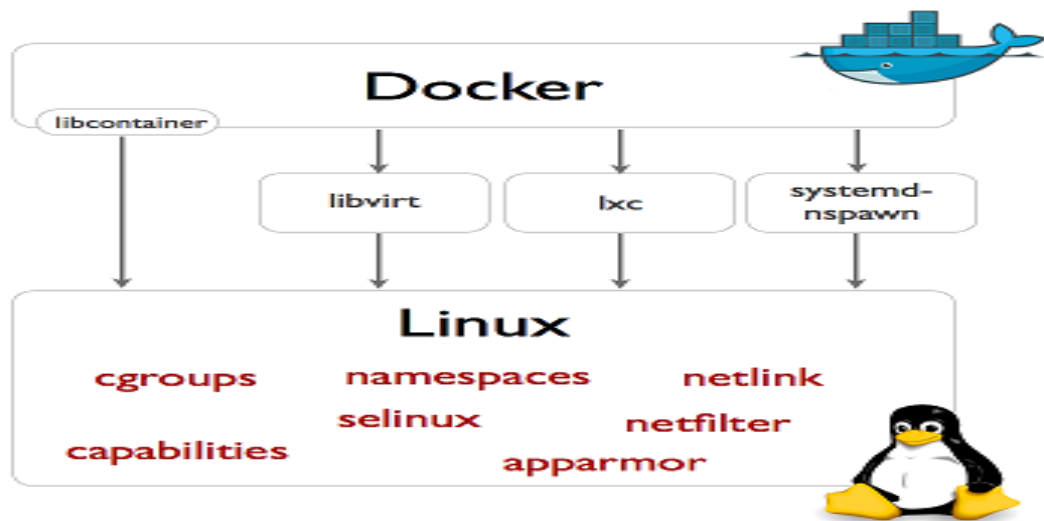
Muita käytettyjä jakeluita ovat SUSE Linux Enterprise Server (SLES), jonka markkinaosuuden on arvioitu olevan 25 %:n luokkaa. Toinen on Ubuntu Long Term Support (LTS), joka on suosittu varsinkin PC:ssä, mutta on siirtymässä yhä vahvemmin palvelinkoneisiin. Oracle Unbreakable Kernel voi olla joissain tapauksissa järkevä valinta, jos yrityksen tietokannat perustuvat Oracleen. [45.]

2.3.8 Kontaineri-virtualisointiarkkitehtuuri

Kontaineri on isäntäkoneen käyttöjärjestelmän Linux-ytimen tason virtualisointia, jonka päällä voidaan suoraan suorittaa sovelluksia ilman ylimääräistä VM-käyttöjärjestelmää (kuva 15).

Tämän ansiosta kontainerivirtualisointi on tehokkaampaa kuin käyttöjärjestelmätason paravirtualisointi, ja se vähentää virtualisointiin tarvittavaa ohjelmiston määrää huomattavasti. Kontainerin voidaan ajatella olevan myös kevennetty VM. Yhä useammat virtualisointitekniikkaa käyttävät yritykset ovat tällä hetkellä siirtymässä Linux Container -virtualisointiin.

Kuvassa 14 esitetty Docker container on tällä hetkellä suosituin Linux Container -ohjelmisto [47].

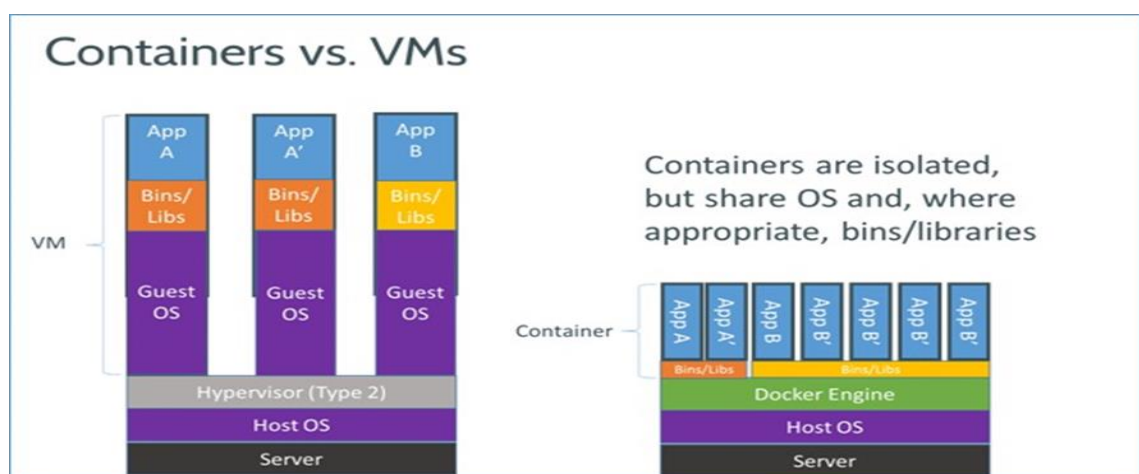


Kuva 14. Docker container -arkkitehtuuri [48].

Docker containerin etuna on, että sen päällä voidaan suorittaa muitakin kuin Linux-käyttöjärjestelmiä, kuten Windowsia. Esimerkiksi Linux LXC Containerin päällä voidaan suorittaa vain Linux-pohjaisia käyttöjärjestelmiä. [48].

Dockerissa QEMU/KVM tuki tulee libvirtin kautta ja Linux Container standardoidun lib-containerin kautta. Tuki on myös Linux Container LXC:lle, joka on ollut ongelmana Dockerille, koska sen rajapinta on muuttunut eri Linux-versioiden välillä. LXC käyttää Linux-rajapintoinaan muun muassa cgroupsia ja namespacesia [48].

Kuvassa 15 havainnollistetaan yleisen Docker containerin ja VM:n eroavaisuuksia.

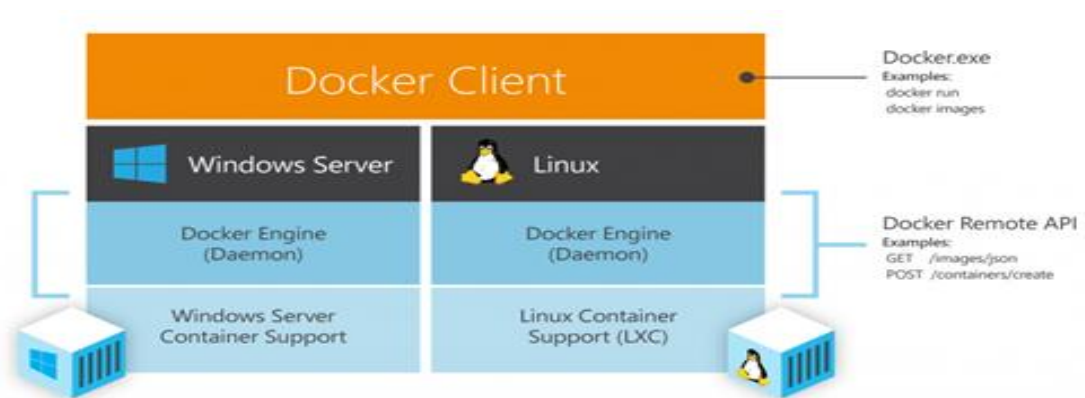


Kuva 15. Vasemmalla 2-tyypin hypervisor ja sen päällä virtuaalikone (VM) ja oikealla Docker container -arkkitehtuuri [49].

Virtuaalikoneet (VM) sisältävät sovelluksen ja tarvittavat binäärit ja kirjastot sekä lisäksi käyttöjärjestelmän.

Docker on kevennetty virtualisointi. Docker container sisältää vain sovelluksen ja tarvittavat binäärit ja kirjastoja ajaa niitä isäntäkäyttöjärjestelmän user spaceissa. Se jakaa isäntäkäyttöjärjestelmän Linux ytimen muiden kontainerien kanssa. Siten se toimii kuten VM eli omaa eristettyjen resurssien jaon, mutta paljon tehokkaammin. [50; 51.]

Docker container -tuki Windowsille on kuvassa 16.



Kuva 16. Docker container -tuki Windows-palvelimelle [53].

Microsoft ilmoitti vuonna 2014, että seuraava Windows Server -versio tukee kehittäjien keskuudessa suosittua Docker container -virtualisointia. Tämä tarkoittaa, että Windows Server on siirrettävissä paikallisista tietokoneympäristöistä back-end-infrastruktuuriin ja pilveen. [53.]

Myös VMware ilmoitti vuonna 2014 yhteistyön aloittamisen Docker containerin kanssa tavoitteinaan etsiä samankaltaisuuksia ja erilaisuuksia VMwaren ja Docker-tuotteiden välillä [54].

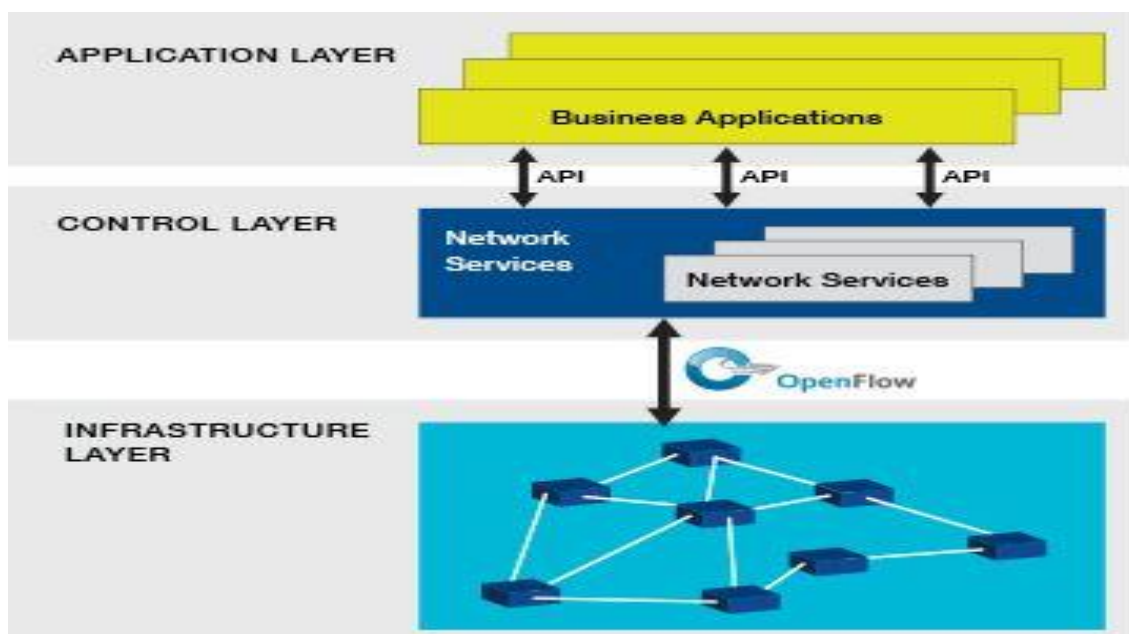
RedHat Enterprise Linux 7 (RHEL 7) ja RedHat Enterprise Linux Atomic, joka perustuu RHEL 7 -järjestelmään, tukevat Docker containeria, joka julkaistiin tammikuussa 2015 [55].

3 SDN/NFV-teknologia

3.1 SDN – ohjelmisto-ohjatut verkot

Ohjelmisto-ohjatut verkot eli Software-defined Networking (SDN) on menetelmä, jossa tietoliikenneverkon ylläpitäjät voivat hallita verkon palveluja esimerkiksi pilvipalvelimesta käsin. Tämä on tehty erottamalla verkkolaitteiden ohjausosa (Control Plane) ja datan siirto-osa (Data Plane) toisistaan. Perinteisesti nämä molemmat ovat sijainneet samassa fyysisessä reitittimessä. [3; 4.]

Kuvassa 17 on esitetty SDN-arkkitehtuuri toteutettuna OpenFlow-kommunikointiprotokollalla.



Kuva 17. Software-defined Networking (SDN) -arkkitehtuuri [79].

Perinteisesti datan siirto-osa on jaettu tasoihin L1, L2 ja L3. Tältä osin kaikki on kunnossa, mutta ohjausosa on vaikeasti hallittavissa, sillä jokainen laite on pitänyt konfiguroida käsin erikseen, kuten reititykset ja pääsyylistat (Access Control List, ACL). SDN:n ohjausosan avulla konfigurointi voidaan hoitaa keskitetysti ja siten, että se on erotettu alla olevasta datan siirto-osasta ja toimii virtualisoidussa tietoliikenneverkossa [4]. Tietoliikenneverkkotoimintojen virtualisointi (Network Function Virtualization, NFV) on tär-

keä osa SDN/NFV:n toiminnassa, ja se käsitellään omassa erillisessä luvussaan tämän SDN-osuuden jälkeen.

Nykyisin käytössä olevien perinteisten tietoliikenneverkkojen ongelmia teleoperaattoreiden näkökulmasta ovat muun muassa.

- vaikeus integroida uutta tekniikkaa
- heikko tehokkuus johtuen tarpeettomista toiminnoista useiden protokollatasojen välillä
- vaikeus mukautua uusiin palveluihin [4].

Teknisestä näkökulmasta tämä vaatii seuraavaa:

- Reitittimien tulee olla ohjelmoitavissa. Pelkkä nykyinen komentotulkki (Command-Line Interpreter, CLI) -rajapinta ei riitä. Ratkaisuna ohjelmoitavuuteen on SDN OpenFlow -rajapinta.
- Tietoliikenneverkkolaitteet ovat jääneet teknisessä kehityksessä jälkeen verrattuna PC-maailmaan, jossa mikropiirien toimittajat (muun muassa AMD, Intel) ja PC-valmistajat ovat eriytyneet. Tietoliikenteessä laitevalmistajat ovat itse suunnitelleet kytkimet ja niihin ohjelmistot. Kytkinten suunnittelu on vaatinut kalliita investointeja kaikilta laitetoimittajilta, ja tämä on tuonut tarpeetonta päällekkäisyyttä, joka on tullut kalliiksi.

Ratkaisuna kustannusongelmaan on käyttää SDN-kytkimien valmistukseen erikoistuneiden toimittajien (muun muassa Broadcom ja Marvell) kytkimiä, joilla jokainen laitetoimittaja voi rakentaa oman laitteiston ja ohjata sitä SDN-ohjelmistoilla. [4.]

Teleoperaattoreilla kestää kymmenen vuotta, ennen kuin uusi ominaisuus saadaan otettua käyttöön. Siinä on erilaisia vaiheita, kuten standardointi, kehitystyö, laitevalmistus, laitteiden hankinta ja käyttöönotto. SDN mahdollistaa uusien ominaisuuksien käyttöönoton paljon nopeammin. [4.]

SDN-tekniikkaa voidaan käyttää laajasti ohjaamaan tietoliikenteen eri solmukohtia muun muassa datakeskuksissa, laajaverkkojen runkoverkoissa (Wide-area backbone),

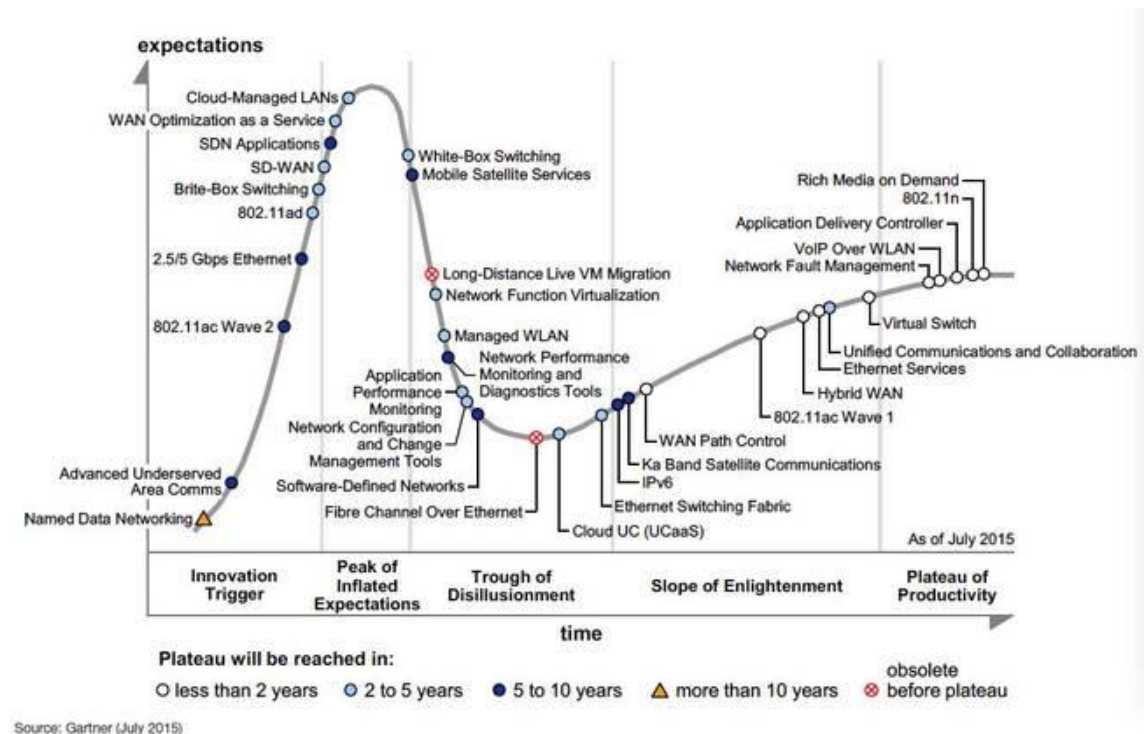
yrittysverkoissa, internetin solmukohdissa (internet exchange points) ja kotiverkoissa. [4.]

Perinteisten tietoliikennelaitetoimittajien näkökulmasta nykyiset reitittimet voidaan pitää yhä suljettuina. Niihin tarvitsee vain tehdä avoin OpenFlow-rajapinta reitittimessä jo olevaan datan ohjaustaulukkoon (Flow table). Näin ollen laitteistopäivitystä ei tarvita. [4.]

Teleoperaattoreille tämä antaa mahdollisuuden käyttää nykyisiä perinteisiä reitittimiä ja edullisia avoimen laitteiston White-Box-kytkimiä hankittuna miltä tahansa laitetoimittajalta. Broadcom ja Marvelin -kytkinkomponenteilla laitevalmistajat voivat rakentaa kohtuukustannuksin SDN-kytkimen. [4.]

3.2 SDN/NFV-teknologian yleistyminen

Kuvassa 18 on Gartnerin "Hype Cycle" -analyysi heinäkuulta 2015. Se kuvaa verkko-tekniologioiden yleistymistä ja mahdollista yleistä käyttöönottoa.

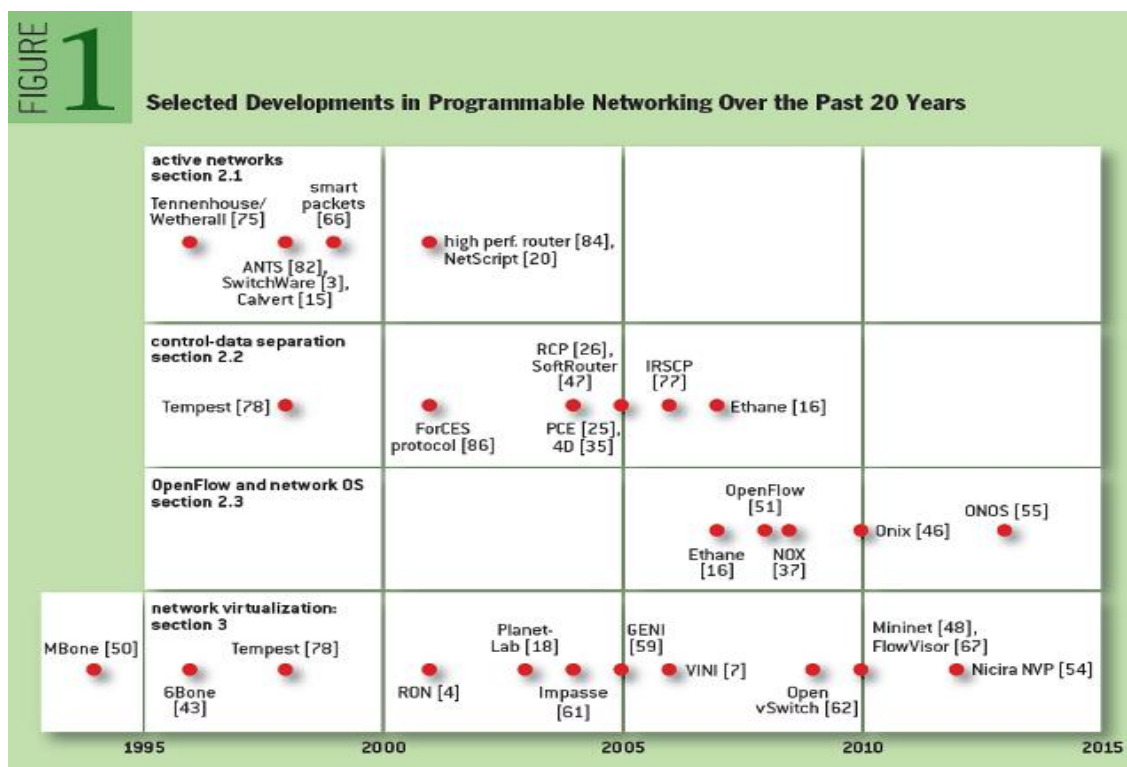


Kuva 18. Gartnerin "Networking and Communications Hype Cycle" -analyysi SDN-, NFV- ja White-Box-tekniologioiden yleistymisestä [58].

Kuvasta voidaan nähdä tässä insinööriyössä käsiteltävien Virtual Switch-, Ethernet Switching Fabric-, SDN-, NFV-, White-Box Switching-, SDN Applications-, SD-WAN- ja Brite-Box Switching -teknologioiden yleistymisen tietoliikenneverkoissa. Esimerkiksi Virtual Switch on jo laajemmassa käytössä esimerkiksi pilvipalvelinympäristöissä ja se on saavuttamassa kahden vuoden sisällä tasaisen vaiheen käyrällä. Tämä on osoitus teknologian menestyksestä ja taloudellisesta tuottavuudesta. Automatisoitu ja yksinkertainen Ethernet Switching Fabric -verkkomalli on korvannut perinteisen puumallisen verkon [77]. Se on vastikään otettu käyttöön datakeskuksissa ja pienentänyt henkilöstökustannuksia. Seuraavaksi tulossa olevan SDN:n osalta suurin ”markkinointihypetys” on jo takanapäin ja ollaan siirtymässä toteutusvaiheen puolelle. Tässä siirtymävaiheessa eli käyrän alimmassa kohdassa teknologiat joko alkavat menestyä ja tulevat yleiseen käyttöön tai sitten eivät. NFV ja muut mainitut teknologiat ovat tulossa SDN:n jälkeen sijoittuen ”hypepiikin” molemmin puolin. Samalta käyrältä voidaan havaita myös muiden tunnettujen ja menestyneiden teknologioiden, kuten IPv6:n, Hybrid WAN:n (Carrier Ethernet ja MPLS) ja 802.11:n (WLAN+MIMO) sijoittuminen käyrän tasaiselle, taloudellisesti kannattavalle nousuvaiheelle. Vuoden 2016 analyysi on juuri julkaistu, mutta saatavana vain maksullisena versiona.

3.3 SDN/NFV:n kehityshistoria

Ohjelmisto-ohjattujen verkkojen kehitys (kuva 19) on viimeisen 20 vuoden aikana ja yleiseen käyttöön valituksi tulleet teknologiat. Tässä insinööriyössä käsiteltyjä teknologioita ovat OpenFlow, NOX, Open vSwitch, Mininet, FlowVisor, Nicira NVP ja ONOS.



Kuva 19. SDN/NFV:n kehityshistoria [59].

SDN-kontrollerin ja kytkimen yhdistävä avoimen lähdekoodin kommunikointiprotokolla OpenFlow ei ole ainoa olemassa oleva protokolla. Se on kuitenkin nykypäivänä vallitseva [4]. Sen takana on suuri joukko yrityksiä. Kahdella isolla kaupallisella yrityksellä VMwarella ja Ciscolla on omat protokollat [60], jotka soveltuvat paremmin niiden omiin tuotteisiin [4].

3.4 SDN-sovelluskohteita

SDN/NFV mahdollistaa nopean tietoliikennearkkitehtuurien kehityssyklin ja uusien innovaatioiden tuottamisen, mikä on tärkein syy siihen, miksi SDN/NFV-teknologiaa kehitetään. Se tuo saman innovaatioiden julkaisunopeuden tietoliikennepuolelle, kuin mitä on totuttu näkemään ohjelmistopuolen kehityksessä. Näitä ovat esimerkiksi IT-palvelujen siirtyminen yritysten omista palvelintiloista pilvipalvelimiin ja Googlen tuotteet. Tärkeimpiin asioihin kuuluvat myös investointi- ja käyttökulujen (CAPEX/ OPEX) alentamistarpeet. Tätä teleoperaattorit ja niiden palveleamat yritysasiakkaat hakevat pärjätäkseen globaalissa kilpailussa. SDN/NFV on laajentumassa eri sovellusalueille ja

kaikkialle. [4.] Lähitulevaisuuden visiona on täysin ohjelmisto-ohjattu infrastruktuuri (Software-Defined Everything, SDx) [1].

SDN:ää käytetään nykypäivänä pilvipalvelimissa lähinnä korvaamaan virtuaalilähiverkko (Virtual Local Area Network, VLAN) -tekniikka joustavammalla ratkaisulla. Esimerkiksi pankkisektorilla, sairaaloissa ja yrityksissä tietoturvalliset verkot ovat tärkeitä ja verkot on perinteisesti luotu VLAN-tekniikalla. Palomuuereilla on hoidettu vain tietyn liikenteen pääsy toiseen VLAN-verkkoon. VLAN:ien luonti on työlästä ja tehdään manuaalisesti komentotulkin komennoilla. Niitä on myös vaikea hallinnoida. SDN:llä vastaavat verkot voidaan luoda automaattisesti. [4.]

SDN:llä päästään käsiksi Internet Protocol (IP) -paketin otsikkotietoihin. Tämän tiedon avulla data ohjataan NFV "middlebox":lle, joka tarkastelee ohjatun paketin sisällä olevaa tiedon sisältöä ja tekee sille määritellyt toimenpiteet. Toimenpiteenä voi olla esimerkiksi palomuuritoiminto. Erilaiset uudet NFV "middlebox" -palvelut ovat iso kasvava alue. [4.]

SDN mahdollistaa pilvipalveluympäristössä tietynlaisen orkestroinnin, joka perinteisellä menetelmällä on ollut enemmän tai vähemmän mahdotonta. Esimerkiksi monitasoiset mallipohja- eli "template"-IT-ympäristöjen luonti on tullut helpommaksi. Perinteisellä tekniikalla on voitu luoda kaksitasoinen mallipohja, jossa on VM-koneita, verkko ja sovellutukset. SDN:llä voidaan luoda kolmitasoinen mallipohja, jossa voi olla NFV "middlebox" -toimintoja, kuten palomuri (Firewall, FW), yhdyskäytävä (gateway, GW), kuormanjako (Load Balancing, LB), web-palvelin, sovellus (applikaatio) ja tietokanta (database, DB). [4.]

3.5 ONF – avoin verkkoyhteisö

Open Networking Foundation (ONF) on voittoa tavoittelematon avoin verkkoyhteisö, jonka perustajajäseniin kuuluvat muun muassa Deutsche Telekom, Facebook, Google, Microsoft, Verizon ja Yahoo. Tavoitteena on edistää verkostoitumista SDN:ään liittyvissä asioissa sekä OpenFlow'n ja vastaavien teknologioiden standardointia [2; 15; 16.]

European Telecommunications Standards Institute (ETSI) on myös aloittanut standardisoinnin [9]. Yleisesti ottaen standardisointi on hidasta ja vaatii usean vuoden työn,

ennen kuin päästään itse toteutukseen. Tämä on aika hidas lähestymistapa luoda SDN/NFV-tuotteita. Nopeampana tapana voi pitää ONF:n lähestymistapaa, jossa avoin yhteisö luo yhteisiä menettelytapoja, jotka voidaan ottaa yhteisön käyttöön [4]. Standardisointitahot määrittelevät pitkällä aikavälillä käytettävät teknologiat, joita mainitut tahot haluavat käyttää. Standardit ovat usein myös sidoksissa tiettyyn maantieteelliseen hallinnointialueeseen.

3.6 OpenFlow-kontrollerit

Yleisimpiä OpenFlow-kontrollereita ovat muun muassa seuraavat:

- NOX-Classic. C++/Python, alkuperäinen OpenFlow kontrolleri, joka ei ole enää käytössä. NOX on Nicira Networksin kehittämä. NOX:a ja OpenFlow'ta kehitettiin rinnakkain. Nicira lahjoitti NOX:n kehittäjäyhteisölle vuonna 2008, jonka jälkeen se on ollut perusta monille kontrollerikehitysprojekteille. [63.]
- NOX. C++, "uusi NOX", hyvin ylläpidetty ja tuettu. Tukee OpenFlow v1.0:aa, sekä 1.1:tä, 1.2:ta ja 1.3:a. Se on tehokas. [63.]
- POX. NOX Python, ainoastaan OpenFlow v1.0:n tuki. Laajasti käytetty, helppo oppia ja kirjoittaa koodia. Heikkoutena on tehottomuus. [64.]
- Ryu. Open source Python-kontrolleri. Tukee OpenFlow v1.0:aa, 1.1:stä, 1.2:sta, 1.3:a, 1.4:ää, 1.5:sta ja Nicira-laajennukset. Toimii OpenStackin kanssa. Tavoitteena on olla "käyttöjärjestelmä" SDN:lle. Heikkoutena on tehottomuus. Ryu on japanilainen. [65; 66.]
- Beacon. Java OpenFlow-kontrolleri, Stanfordin yliopisto.
- Floodlight. Open source Java and Web based GUI, joka perustuu Beacon-kontrolleriin. Tukee OpenFlow v1.0:aa. Big Switch Networksin ylläpitämä. On hyvin dokumentoitu. REST API -tuki. Tuotetasoinen tehokkuus ja tuki. Tukee OpenStack/Multi-tenant-pilvipalvelimia. Monimutkainen ja vaikeahko oppia. [67.]

- **LoxiGen.** Tukee monia ohjelmointikieliä, kuten C:tä (kutsutaan "loci"). Tulossa on Python- ja Java-tuki. Perustuu Floodlight-kontrolleriin. Tavoitteena on päivittyä OpenFlow'n eri versioiden tahdissa ja antaa tuki eri OpenFlow-versioille ja ohjelmointikielille. Tukee OpenFlow v1.0–v1.3.1:tä. Ohjelmointikielitet: C-tuki Indigolle, Java-tuki Floodlightille, Python OpenFlow-testaukseen, Lua Wiresharkille. Big Switch Networksin ylläpitämä. [68.]
- **OpenDaylight.** Java and Web based GUI, NFV, myös non-OpenFlow, perustuu Beacon-kontrolleriin. Laitevalmistajien tukema alusta. Laajennettavissa oleva ja perustuu avoimen lähdekoodin ohjelmistoon. Siihen on saatavissa laitevalmistajien ohjelmistoja. Laajaan **OpenDaylight Open SDN -ekosysteemiin** on liitetty monia muita avoimia ja kaupallisia kontrolleita, joiden tulisi kyetä toimimaan samassa verkossa keskenään. OpenStack-tuki. Laaja "Southbound" API -tuki kytkimeen päin johtuen useista laitevalmistajista. Se on hyvin monimutkainen ja vaikea oppia. [69.]
- **HP VAN.** Perustuu avoimeen lähdekoodiin ja Beacon-kontrolleriin.
- **OpenContrail.** Juniper NFV. [62.]

Lisäksi on joukko muita kehitteillä olevia kontrollereita ja ohjelmointikieliä, kuten

- Kinetic: (= Pyretic + Frenetic) [4]
- Pyretic: (= Frenetic + Python), pohjalla on POX-kontrolleri, joka voidaan tarvittaessa vaihtaa toiseen [4]
- Frenetic-OCaml: (= Frenetic + OCaml), Ohjelmointikieli [4]
- Frenetic: Princetonin ja Cornellin yliopistot, Yhdysvallat, Ohjelmointikieli [4]
- Procera: Yalen, Georgian ja Marylandin yliopistot, Yhdysvallat [4]
- RouteFlow
- Trema
- Maestro: Java-pohjainen, Tutkimuskäyttöön
- OpenMul
- OpenIRIS

sekä uusimpina

- **ONOS.** Open Network Operating System. Open source Java and Web based GUI. Verkonhallintaohjelmisto, johon voidaan liittää mikä tahansa kontrolleri. Vakioasetuksena Niciran kehittämä kontrolleri. ON.lab, ONOS project ja Internet2 ovat rakentaneet Yhdysvaltojen, Brasilian, Chilen ja Euroopan laajuisen SDN-verkon muun muassa yliopistojen tutkimus- ja koulutuskäyttöön [72]. Euroopan tutkimus- ja koulutusorganisaatio GEANT:iin kuuluvat myös Pohjoismaat NORDUnetin ja Suomi Funetin kautta. Funet on Suomen yliopisto-, korkeakoulu- ja hallinto-organisaatioiden käyttämä tietoverkko [72]. GEANT:n verkko perustuu ONOSin ohella Infineran optiseen pakettitekologiaan ja Corsan ohjelmoitaviin kytkimiin ja reitittämiin [73]. ONOSin ohjelmistoversiojulkaisun (Goldeneye 1.6.0) esitysmateriaalissa mainitaan ”SDX-L2/L3 ONOS Bandwidth on Demand Self serve portal” -ominaisuuden ”GEANT (Europe) Infinera” olevan valmis Q4/2016 [74].
- **OFNIC.** C++ ja Python. Perustuu Niciran kehittämään NOX-kontrolleriin. EU:n osarahoittaman FIWAREn ylläpitämä. Kehittäjinä on muun muassa eurooppalaisia laitevalmistajia ja yliopistoja. [156.]

Kontrollorit eroavat toisistaan käyttötarkoituksen, tietoturvan ja ohjattavan kytkimen välittämän datan välitysnopeuden suhteen.

3.7 OpenFlow-kommunikointiprotokolla

OpenFlow on kommunikointiprotokolla, jolla yhdistetään keskitetty ohjausosa (Control Plane) ja datan siirto-osa (Data Plane) toisiinsa [78; 79].

Muita vastaavia kommunikointiprotokollia ovat Open vSwitch Database Management Protocol (OVSDB), A Data Modelling Language for NETCONF (YANG), Network Configuration Protocol (NETCONF) ja Simple Network Management Protocol (SNMP) [4]. Myös Internet Engineering Task Force (IETF) -työryhmä Interface to the Routing System (i2rs) on kehittänyt SDN-standardin RFC-7426, joka mahdollistaa SDN-kontrolleriohjauksen hyödyntäen perinteisiä protokollia, kuten Open Shortest Path First (OSPF), Multiprotocol Label Switching (MPLS), Border Gateway Protocol (BGP) ja Intermediate System to Intermediate System (IS-IS). Tässä standardissa reitityspäätök-

set on jaettu siten, että se sallii edelleen hajautetun reitityksen. Se sallii myös SDN-kontrollerisovelluksen tehdä muutoksia reititykseen. [10.]

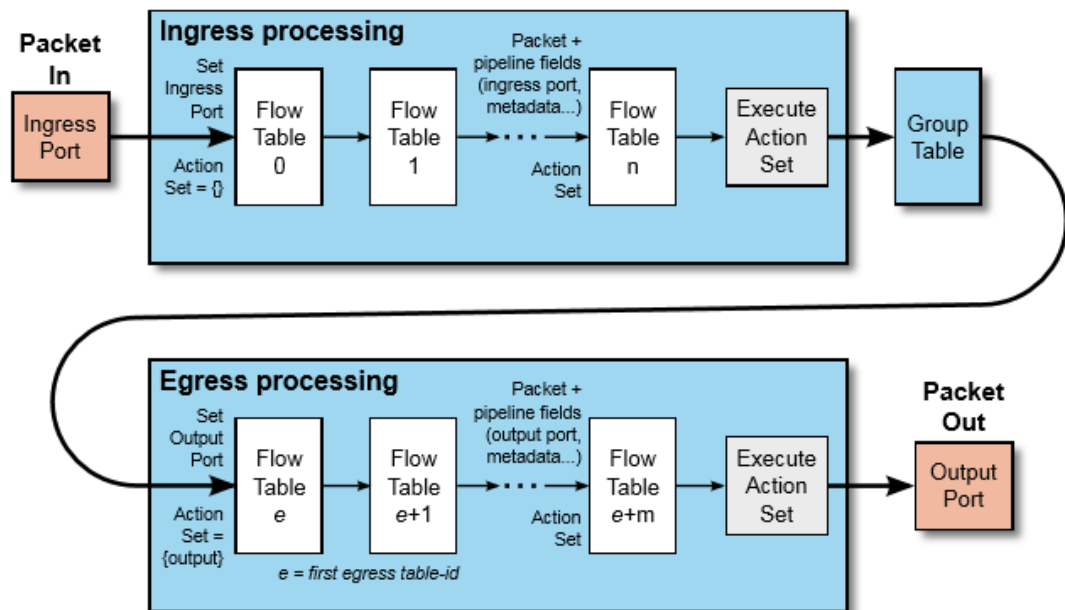
OpenFlow-kommunikointiprotokollassa suositellaan käytettäväksi vähintään kahta ohjausosaa: ensimmäinen pääasialliseksi ohjausosaksi ja toinen varalla. Ohjausosan ja datan siirto-osan erottaminen toisistaan mahdollistaa reititysprotokollien ja pääsylistojen Access Control List (ACL) käytön näiden välillä, joten ne voivat sijaita kaukana toisistaan. Avoin protokolla OpenFlow antaa laitetoimittajille mahdollisuuden käyttää omia liitäntöjään ja skriptikieliä tietoliikenneverkon kytkinten kauko-ohjaukseen. [4.]

OpenFlow-versioita ovat versiot 1.0–1.5. Uuden version mukana tulee aina uusia ominaisuuksia. Esimerkiksi 1.1:n mukana tulee muun muassa MPLS- ja VLAN-tuki. Versiossa 1.2 puolestaan tulee muun muassa IPv6-tuki. Versiossa 1.3 tulee muun muassa lisää IPv6- ja MPLS-tukea. Versiossa 1.4 tulee muun muassa OpenFlow-viestien niputtaminen (bundles), mikä helpottaa ohjausviestien synkronointia useiden ohjattavien kytkinten suuntaan. [80.]

On huomattava, että aina kun laitevalmistaja haluaa lisätä uuden OpenFlow-versiotuen, muutoksia tarvitsee tehdä myös itse kytkimeen eli datan siirto-osaan ja kontrolleriohjelmistoon. Tämän vuoksi kytkimen laitteisto on sidottuna kontrolleriin ja kytkin on niin sanottu Black-Box. Tätä ongelmaa poistaa uusiin Application Specific Integrated Circuit (ASIC) -kehitys. Julkistettuja uusia arkkitehtuureja ovat Intel FlexPipe, Reconfigurable Match Tables (RMT) [SIGCOMM 2013] ja Cisco Doppler. Ohjelmointi näihin uusiin ASIC-ympäristöihin on vaikeaa, ja ne tarjoavat vain liityntärajapinnan kytkinlaitteistoon, eli ne eivät ole täysin avoimia White-Box-kytkimiä. [4.]

3.7.1 OpenFlow-vuotaulukko

Kuvassa 20 esitetään paketin siirtymistä sisääntuloportista (Ingress) ulostuloportiin (Output). Vuotaulukko (Flow table) sijaitsee datan siirto (Data Plane) -osassa ja suorittaa kytkimeen saapuvien pakettien (Packet In) ohjauksen siten, että paketin tunnistetietoja (header) verrataan taulukon tietoihin, ja jos tiedot ovat samoja, toimitaan (Action) kyseisen tiedon mukaisesti. Paketti joko ohjataan eteenpäin kytkimen ulostuloportista (Output) tai se pudotetaan (Forward/Drop). Jos paketin vertailutiedot eivät ole samoja, paketti ohjataan OpenFlow-kanavaa pitkin kontrollerille ohjaustiedon saamiseksi.



Kuva 20. Paketin kulkeminen Input-portista Output-porttiin uusimman ONF OpenFlow -versio 1.5.1:n kuvauksen mukaisesti [81].

Seuraavaksi käydään syvällisemmin läpi OpenFlow 1.0- ja 1.3-versioiden toimintaa, koska ne ovat toiminnoiltaan yksinkertaisemmat seuraaviin versioihin verrattuna. OpenFlow v1.0 tukee seuraavia paketin tunnistetietoja, joita on kaksitoista: Switch Port, MAC src, MAC dst, Eth type, VLAN ID, IP src, IP dst, IP prot, TCP sport, TCP dport sekä lisäksi VLAN priority ja Type of Service (ToS).

OpenFlow v1.0:n toiminta Forward/Drop- ja Modify/Enqueue-tilanteissa:

Actions (Forward/Drop):

Forward:

- ALL: Lähetä paketti ulos jokaisesta liittymästä paitsi ei siitä mistä se on tullut
- CONTROLLER: Kapseloi paketti ja lähetä kontrollerille
- LOCAL: Lähetä kytkimen omaan verkkopinoon
- TABLE: Lähetä paketti FlowTable tiedon mukaisesti. Vain (Packet-out)-viestit
- IN PORT: Lähetä paketti samasta portista mistä se on tullutkin

- Valinnainen vaihtoehto: Lähetä paketti normaalisti ja käyttäen "spanning tree"-protokollaa, jolla poistetaan muun muassa L2-tason data silmukat.

Drop: Jos paketin tunnistetieto ei sisältänyt mitään edellisistä FORWARD toiminnoista, niin paketit pudotetaan.

Näiden toimintojen lisäksi on myös valinnaisia toimintoja (Optional Action).

Optional Actions (Modify/Enqueue):

Modify: Mahdollisuus muuttaa paketin header-tietoja

- Aseta VLAN ID, prioriteetti, ym.
- Aseta destination IP-address. Tätä käytetään esimerkiksi (Wide Area Network, WAN)-verkon kuormanjako (Load Balancing, LB)-toiminnoissa.

Enqueue: Lähetä paketti jonoon, joka on liitetty lähtevään porttiin. Tätä voidaan käyttää QoS:ssä tai "Traffic Shaping"-toiminnoissa.

OpenFlow v1.3:n toiminta tuo mukanaan seuraavia parannuksia sisään tulevan paketin (Packet In) tunnistetietojen käsittelyyn, ennen kuin paketti lähetetään kytkimestä ulos (Packet Out):

Action set: Jokaiselle paketille voidaan suorittaa joukko toimintoja (Actions)

Group: Taulukko, jossa on lista Action set -toimintoja.

- Suorita kaikki Action set -toiminnot. Tämä on käyttökelpoinen, jos paketti halutaan lähettää "multicastina". Yksi paketti on kloonattu ryhmän (Group) jokaiseen Action set:iin.
- Epäsuorat ryhmät (Group). Käyttökelpoinen, kun yksi Action set-toiminto halutaan osoittaa monelle sisään tulevalle paketille. Samanlainen kuin RCP-optimointi [4].

Taulukkoja voi olla useita peräkkäin, ja näistä jokainen voi muuttaa paketin tunnistetietoja ja lisätä Action set -toimintoja.

Lopuksi Action set -toiminnot suoritetaan paketille ja lähetetään kytkimen ulosmeno (Output) -portista ulos (Packet Out).

Esimerkkejä lisätoiminnoista, joita yllä olevilla toiminnoilla voidaan suorittaa:

- TTL: Vähennä TTL-laskuria, kopioi TTL-paketin tunnistetietoihin
- MPLS: Lisää pakettiin MPLS push -toiminta
- QoS: Lisää pakettiin QoS-toiminta esimerkiksi set_queue

Lisäksi on muita asioita, joita OpenFlow'lla voidaan tehdä:

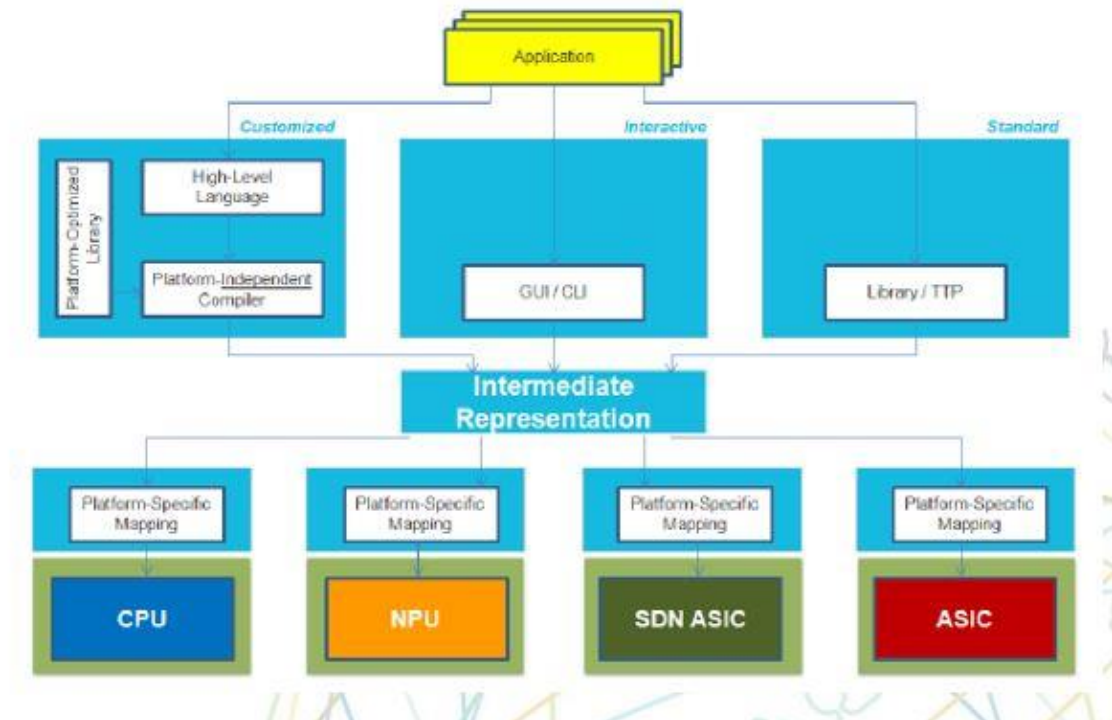
- Voidaan mitata ja monitoroida liikennettä.
- Kontrolliviestejä voidaan vastaanottaa useista kontrolleereista yhtä aikaa. Kontrollikanava voidaan salata.

Lisätietoa OpenFlow-versioista on saatavissa ONF:n spesifikaatioista [80].

3.7.2 OpenFlow 2.0 -protokolla - P4 ja POF Framework

Vielä julkaisematon ja tällä hetkellä varsin keskeneräinen ONF:n määrittelemä OpenFlow'n versio 2.0 tulee poistamaan luvussa 3.7 kuvatut Black-Box-ongelmat [87, 4]. Siinä voidaan käyttää esimerkiksi kehityksen alla olevaa korkean tason Programming Protocol -independent Packet Processors (P4)- tai muuta vastaavaa ohjelmointikieltä datan siirto-osan (Data plane) ohjelmointiin [4; 92; 93; 94]. Huaweiin julkaisema "Protocol Oblivious Forwarding POF framework" tarjoaa kokonaisratkaisun siirtymisessä Black-Box-ympäristöstä White-Box-ympäristöön. POF:ssa voidaan käyttää esimerkiksi P4 (myös C tai Java) -ohjelmointikieltä sekä komentotulkin tai graafisenkäyttöliittymän - rajapintaa POF-kontrollerin ohjaukseen [88; 89; 90; 91.]

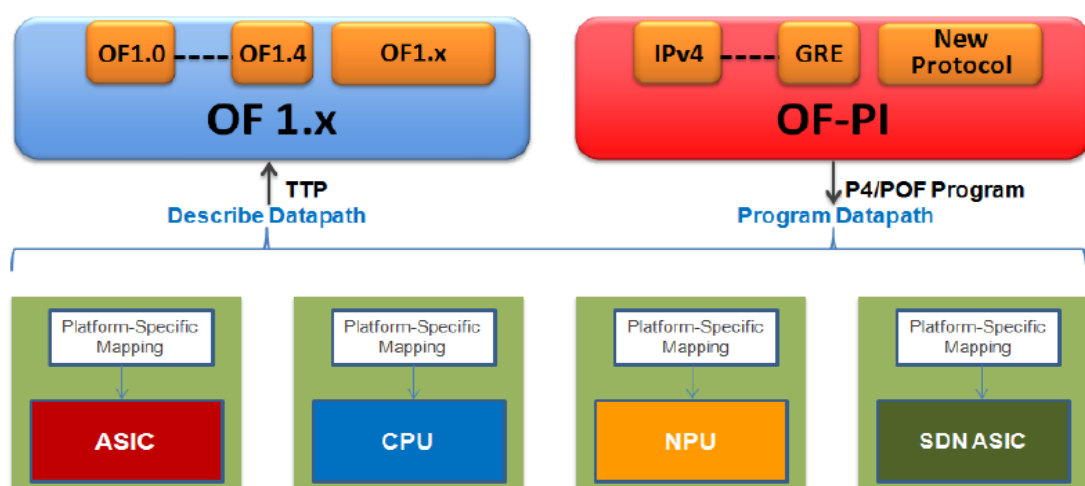
Kaavio POF-ohjelmoinnista on esitetty kuvassa 21.



Kuva 21. POF Programming model [88].

Kuvassa 21 näkyvä Intermediate Representation -lohko on myös ONF:ssä työn alla. Yhtenä vaihtoehtona on esitetty NetASM:a, mutta myös tämä asia on vielä avoinna.

Kuva 22 selventää kuvassa 21 esitettyjen haarojen TTP (Table Type Patterns) ja OF-PI suhdetta.



Kuva 22. TTL ja OF-PI [88].

TTP pitää sisällään Open Flow'n eri versiot ja OF-PI puolestaan P4/POF-ohjelmointi-
osuuden. Näin turvataan nykyistä OpenFlow-protokollaa käyttävien SDN Black-Box ja
suunnitteilla olevien SDN White-Box- ja Brite-Box-kytkimien kehitys.

P4-kääntäjän POF Frameworkin on tulevaisuudessa tuettava erilaisia kytkinarkkiteh-
tuureja, kuten perinteisiä ASIC-kytkimiä, Flexible Match+Action ASIC -kytkimiä (Intel
FlexPipe, Cisco Doppler, Xpliant, jne.), ohjelmoitavia Field Programmable Gate Array,
(FPGA) -kytkimiä (Xilinx, Altera, Corsica, jne.), Network Processing Unit (NPU) -kytkimiä
(Ezchip, Netronome, jne.) ja ohjelmistopohjaisia prosessori (Central Processing Unit,
CPU) -kytkimiä (OVS, jne.). Tämä kehitys vie vielä paljon aikaa [4]. P4-ohjelmointiin voi
tutustua Courseran kurssilla [4] ja POF Framework -kontrolli- ja -kytkinympäristöön
Huawein tutkimushankkeen kautta. [88.]

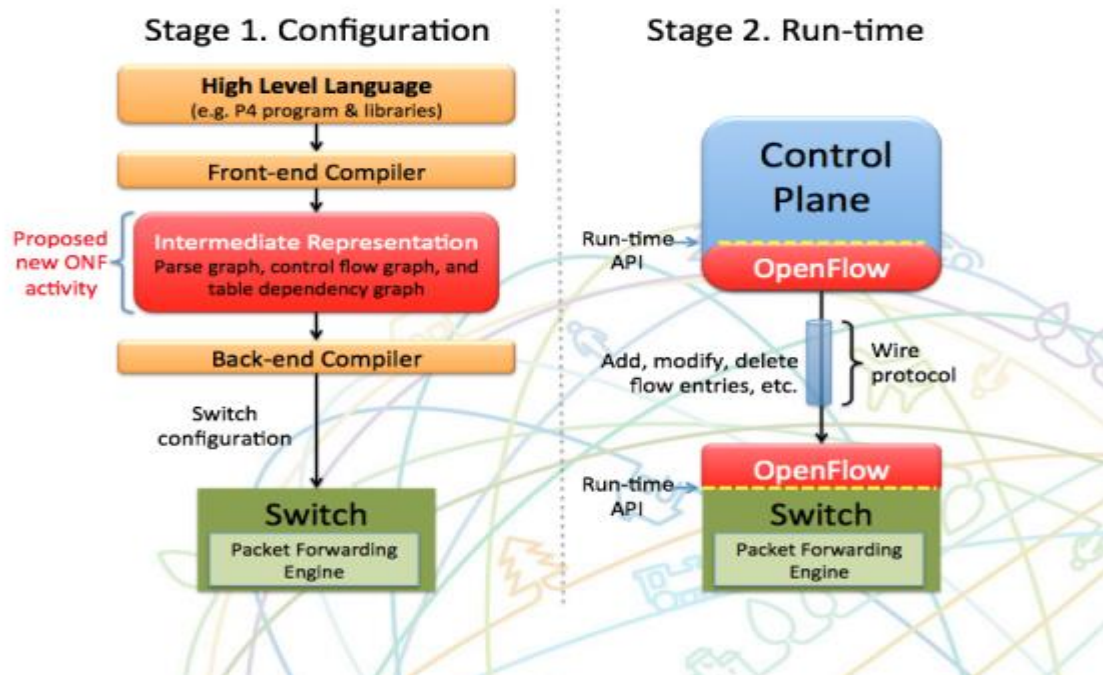
Edellä mainitut arkkitehtuurit eroavat toisistaan lähinnä tehokkuuden ja joustavuuden
suhteen. Tehosuhteet ovat ASIC (/1), FPGA (/10) ja CPU (/100). Tehokkain on ASIC ja
joustavin on CPU-arkkitehtuuri ohjelmoitavuutensa ansioista. [4.]

Markkinoilla olisi kysyntää kytkimelle, jossa olisi yhdistettynä FPGA-pohjainen ohjel-
moitava kytkin ja yleinen "x86 forwarding pipeline". Tämä antaisi mahdollisuuden
FPGAn tehokkuuteen ja x86:n joustavuuteen. [4.]

Nykyiset verkkokytkimet on koodattu tietoliikenneinsinöörin käyttämällä C- tai C++-
kielillä, mutta verkkoinsinöörit, jotka käyttävät ja konfiguroivat näitä verkkokytkimiä,
ymmärtävät hyvin verkkoprotokollat ja niiden toiminnot. He tarvitsevat itselleen sopi-
vamman ja ymmärrettävämmän kielen. P4-ohjelmointikieli on suunniteltu tähän tarpee-
seen. Sillä pystytään koodaamalla konfiguroimaan verkon reititykset ja myös datan
siirto-osan (Data Plane) alimmat tasot. [4.]

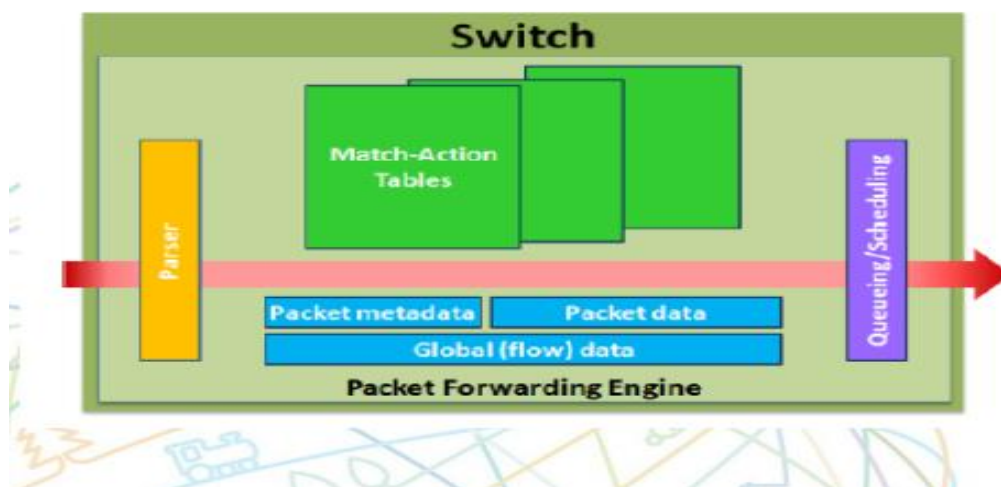
IR (Intermediate Representation) -tuki tulee P4:ään 2–3 vuoden kuluttua. Ensin tarvi-
taan verkkoinsinööreille P4-ohjelmointikieli. [4.]

Protokollasta riippumattoman ohjelmoinnin vaiheet (Protocol-Independent Program-
ming Stages) näkyvät kuvassa 23.



Kuva 23. Protocol-Independent Programmingin eri vaiheet [88].

Vaihe 1:ssä käännetään P4:llä tai muulla, kuten C- tai Java-ohjelmointikielellä, muodostettu ohjelma ja ladataan se kentällä olevaan kytkimeen sekä kontrolleriin. Vaihe 2:ssa käynnistetään ladatut ohjelmat. Kytkinlohko (Switch) pitää sisällään kuvan 24 toiminnallisuudet ja (Parser):lla luetaan paketin osoitetieto, (Match-Action Tables):ssa on reititysohjeet, lopuksi paketit ohjataan jonoihin.



Kuva 24. Tiivistelmä Forwarding Element Modelista [88].

Nämä toimenpiteet voidaan tehdä aiemmin mainitulla NetASM-ohjelmointikielinympäristöllä. NetASMilla muodostetaan rajapinta kytkimen OpenFlow ja muille ohjausprotokollille. NetASM on protokollariippumaton, ja sillä voidaan kääntää muun muassa seuraavia ohjelmointikieliä: NetKat, P4, OpenState, OpenFlow 1.x ja Flowlog. NetASMin periaatteena on, että se tukee kaikkia kytkimen ohjelmointikieliä, protokollia ja itse kytkinalustoja, kuten FPGA, Click, NPU, GPU, Open vSwitch ja Open Data Plane. [4.]

Näiden tulevaisuuden menetelmien ansiosta ollaan riippumattomia käytettävistä protokollista ja kytkimien laitteistosta. Tämän ansiosta kentällä olevat White-Box-kytkimet voidaan päivittää uudelleen tarvitsematta vaihtaa niitä uusiin. Tämä vähentää operaattoreiden investointikustannuksia huomattavasti. Kehitystä tukee myös SDN-tietoverkkojen kytkintoimittajien kiinnostus lisätä kytkimiinsä protokollia, kuten NVGRE ja VXLAN.

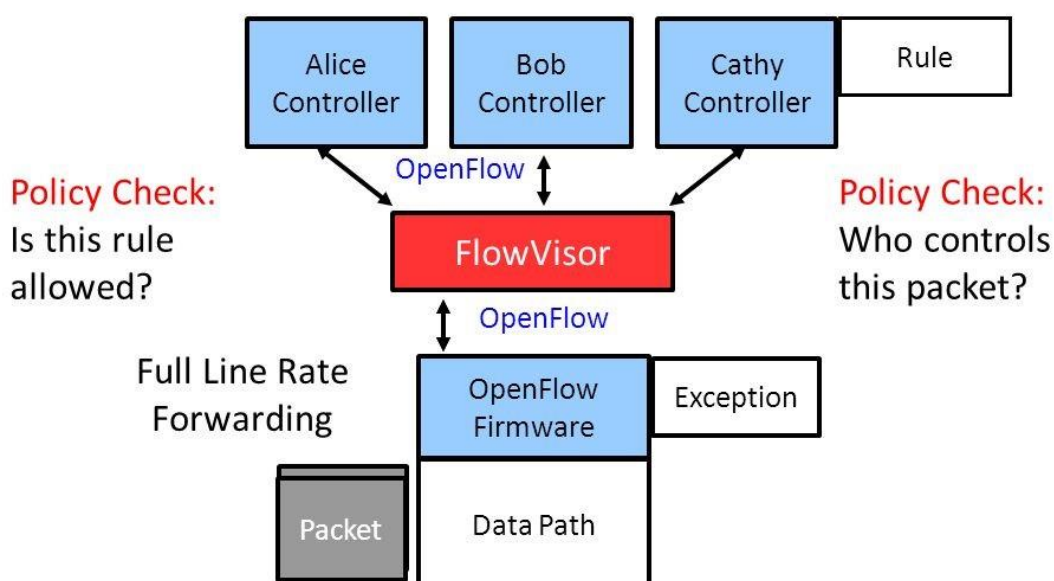
ONF:n työryhmä (Protocol Independent Forwarding, PIF) on koonnut yhteen kahden ryhmittymän (POF ja P4) työn tulokset ja julkaissut 5.9.2014 ensimmäisen ONF-spesifikaation "OF-PI: A Protocol Independent Layer for OpenFlow v1-1", jonka on tarkoitus olla pohjana tulevalle OpenFlow'n versio 2.0:lle [83; 84]. Huaweiin Protocol Oblivious Forwarding Framework (POF) on esitetty Barcelonassa SDN World 2013 -tapahtumassa [88].

Huaweiin esittämä POF on liitetty ONF OF-PI -spesifikaatioon. POF on avoimen ohjelmiston projekti ja ohjelmistojen julkaisut löytyvät Huaweiin verkkosivuilta [91]. Ohjelmistojulkaisut ovat päättyneet vuonna 2013, ja ne ovat jääneet odottamaan ONF-spesifikaation valmistumista, ennen kuin jatkavat julkaisujaan.

POF-kontrolleri on BSD/Apache-lisensioitu Java-pohjainen OpenFlow-kontrolleri. Se tarjoaa graafisen käyttöliittymän kytkimen kontrollointiin ja konfigurointiin. POF-kytkin on BSD-lisensioitu ja Linux C -kieleen perustuva virtuaalikytkin.

3.8 OpenFlow-yhteyden virtualisointi

Myös SDN-kontrollerin ja kytkimen yhdistävä OpenFlow-yhteys voidaan virtualisoida kuvan 25 esittämällä tavalla.



Kuva 25. FlowVisor SDN -virtualisoinnin arkkitehtuuri [4].

Tämä tarkoittaa, että kytkimen ja kontrollerin väliin tulee uusi välikerros (Slicing Layer), joka jakaa kytkimen L1–L4 tason liikenteen usealle kontrollerille. Jako voidaan suorittaa L1-portin, L2-ethernet src/dst:n, L3 IP src/dst:n, L4 Transmission Control Protocol (TCP)/User Datagram Protocol (UDP)-portin tai Internet Control Message Protocol (ICMP)-koodi/tyypin mukaan. Tällöin tietyntyyppinen liikenne virtualisoidaan ja siten saadaan liikenne erotettua varsinaisesta käytössä olevasta tuotantoverkosta. Näin tuotekehittäjät voivat kehittää liikennetyypin esimerkiksi Voice Over Internet Protocol (VoIP)-, Hypertext Transfer Protocol (HTTP)- ja pelialan tuotteita virtualisoidussa SDN-verkossa. Kuvatuslainen virtualisointi voidaan tehdä esimerkiksi FlowVisor SDN -kontrolleriohjelmalla. [4.]

3.9 Mininet-verkkoemulaattoriympäristö

OpenFlow- ja SDN-toimintaa voidaan helposti havainnollistaa Mininet-emulaattorilla PC-ympäristössä [96]. Mininet-verkkoemulaattoria varten tulee asentaa PC:lle Oracle VM VirtualBox. Mininet-PC-ympäristöllä voidaan luoda realistinen virtuaaliverkko ilman fyysisiä White-Box-kytkimiä ja ohjausosaa [95]. Mininet on käyttökelpoinen ympäristö OpenFlow- ja SDN-tuotekehitykseen, opetukseen ja tutkimukseen.

Mininet on kevyt ympäristö, jota käytetään laajasti muun muassa Silicon Valleyn verkko- ja pilvipalvelinyrityksissä Yhdysvalloissa. Sitä käytetään esimerkiksi erityyppisten SDN-kontrollereiden ja SDN-sovelluksien tuotekehityksessä ja -testauksessa sekä virtuaaliverkon järjestelmätestauksessa, niin sanottu päästä päähän testauksessa. Suurten klusteri-ympäristöjen toteutukseen on kehitetty Maxinet-ympäristö. Mininetin etuna on helppo parametrisointi esimerkiksi verkkotopologian vaihdoksissa testauksen aikana. Yksittäisiä komentoja voidaan antaa esimerkiksi Extensible Markup Language (XML)- ja Java Script Object Notation (JSON)- ym. API-rajapintojen kautta, mutta automaatiotestit kannattaa toteuttaa esimerkiksi Python-ohjelmointikielellä.

Mininetin toiminnan periaatteena on suorittaa sitä yhden PC:n sisällä ja käyttää sen omia Linux-käyttöjärjestelmäohjelmiston prosesseja siten, että niihin muodostetaan virtuaalinen Ethernet-liitäntä ja IP-osoite sekä voidaan nimetä käyttäjät (users) ja ryhmät (groups). Näin esimerkiksi päätelaitteita tai kytkimiä voidaan helposti lisätä Mininet-verkkoon muodostamalla lisää kuvatuslaisia Linux-prosesseja. Mininet-ympäristöön voidaan liittää myös SDN-kontrolleri ohjaamaan kytkimen toimintaa. Jos tuotekehityksen laboratorioverkkoympäristössä on tarve käyttää esimerkiksi erillisiä Windows- tai BSD-käyttöjärjestelmän koneita osana laboratorioverkkoa, silloin tulee rakentaa järjestelmälle todellinen virtuaaliympäristö. MiniEdit on graafinen ympäristö (GUI) Mininetille. Toinen samantyyppinen verkkoemulaattori kuin Mininet on Common Open Research Emulator (CORE). [4.]

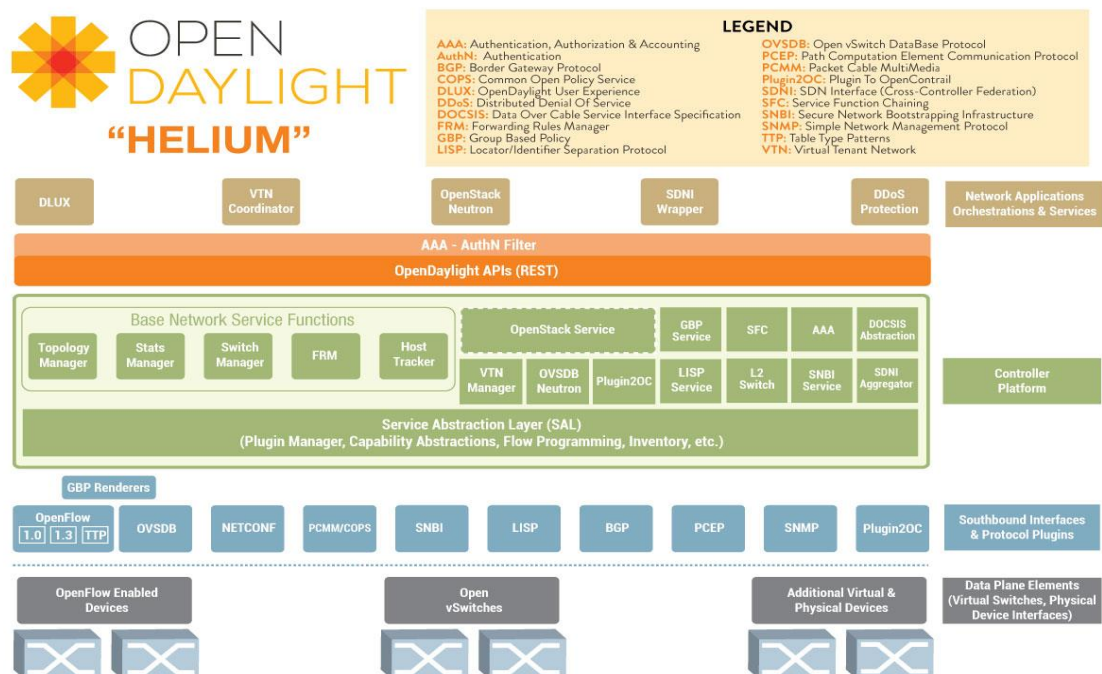
Vagrant on VM-koneen hallintatyökaluohjelmisto, jolla käynnistetään VM-kone ja siinä suoritettavat ohjelmistot, kuten Mininet, konfigurointitiedoston Vagrantfile-määrittysten mukaisesti. Vagrant-ohjelmaa on käytetty myös Princetonin yliopiston SDN-kurssiympäristön jakamiseksi ja pystyttämiseksi opiskelijoiden omille PC:lle. Vagrant-ympäristön käyttöönotto kuvataan myöhemmin tarkemmin luvussa 5.3.3 SDN/NFV-ympäristön toteutus.

3.10 Avoimet SDN-kontrollerit

Avoimia SDN-kontrollereita ovat muun muassa OpenDaylight, Open Contrail, Floodlight, Ryu ja ONOS, joka on uusin tulokas [61].

Lupaavimpia avoimia kontrollereita ovat OpenDaylight ja ONOS. OpenDaylight-alustalla asiakkaat ja laitetoimittajat voivat innovoida ja tehdä yhteistyötä sekä kaupallistaa SDN- ja NFV-teknologioihin perustuvia ratkaisuja. OpenDaylightin kehittäjinä ovat lähinnä suurten suljettujen tietoliikennejärjestelmien laitevalmistajat ja ohjelmistoyritykset. Sen kehittämisessä ovat mukana esimerkiksi Brocade (ostettiin Verizonin vuonna 2012), Cisco, Big Switch, Juniper, Ericsson, IBM, HP, Dell, Microsoft, Intel, Citrix, RedHat, VMware ja Oracle [70].

Kuvassa 26 esitetty OpenDaylight-ohjelmisto Helium sisältää liitettäviä komponentteja, kuten kontrolleri, erilaiset rajapinnat, protokollalaajennukset ja sovellukset, joiden ansiosta OpenDaylight on muutettavissa yritysten tarpeisiin sopivaksi.



Kuva 26. OpenDaylight SDN -ohjelmistoarkkitehtuuri [69].

ONOS ja OpenDaylight on koettu kilpaileviksi kontrollereiksi. OpenDaylight-alustaan on liitetty vastikään myös ONOS [71].

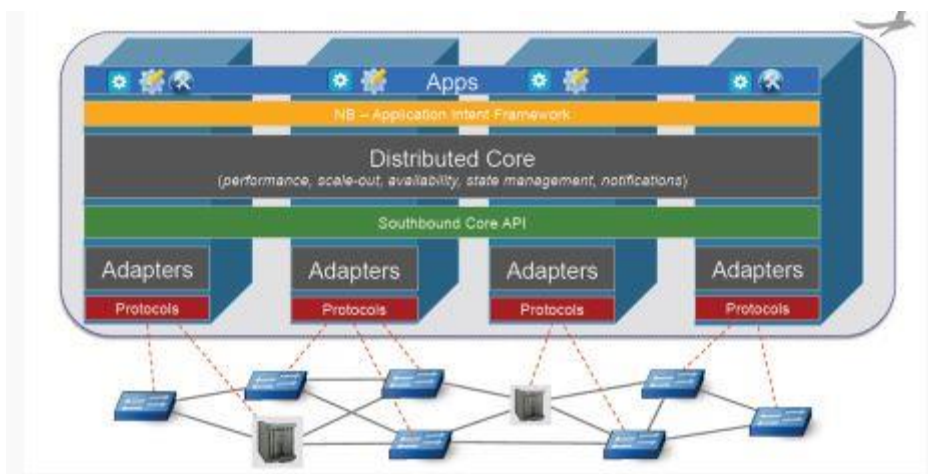
ONOS (kuva 27) on operaattoreiden itsensä näkökulmasta ja niiden tarpeisiin kehitetty verkonhallinta- ja kontrollerialusta, jolla vältetään lukkiutumasta laitetoimittajan tuotteisiin ja liikemalleihin (vrt. OpenDaylight). [71.]



Kuva 27. ONOS Open Network Operating System [75].

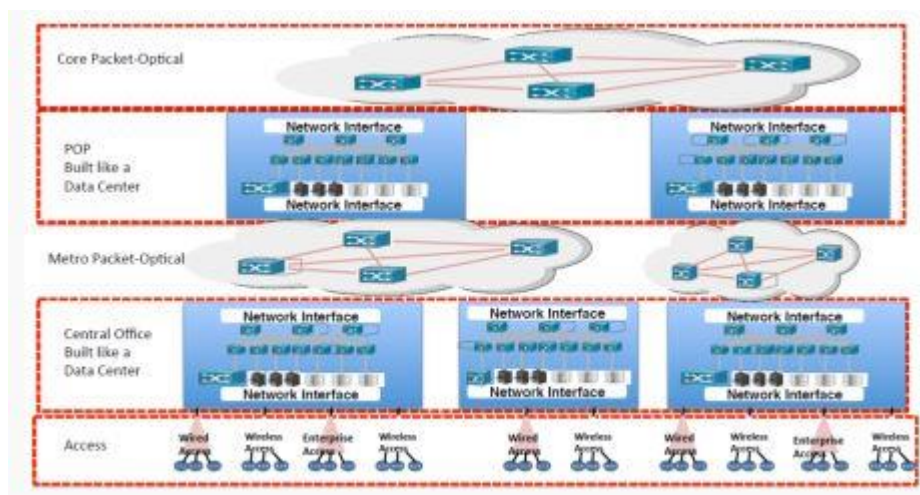
ONOS:n kehittäjänä on ollut ON.Lab (Open Networking Lab) ja tukijoina muun muassa operaattorit AT&T, NTT, SK Telecom sekä ONF-järjestön tukijat, kuten Google, Facebook, Yahoo, Verizon ja Microsoft [71]. Myös laitevalmistajia on nopeasti liittynyt mukaan, esimerkiksi Ericsson, Nokia, Ciena, Cisco, Huawei, Intel, NEC ja Fujitsu.

ONOS on hajautettu SDN-kontrolleri (kuva 28). Se on tällä hetkellä lähinnä operaattoreiden optisiin Wide Area Network (WAN) -verkkoihin tarkoitettu, mutta suunnitelmassa on kehittää sitä käytettäväksi myös pilvipalvelimissa ja yrityksissä. ONOS:n soveltuvuudesta eri käyttötarkoituksiin on meneillään useita käyttökokeiluja (Use Case), kuten Segment routing, Network Function As A Service (NFaaS), Multi-layer SDN ja SDN-IP. [75; 76.]



Kuva 28. ONOS:n hajautettu arkkitehtuuri [75].

ONOS on keskittynyt käyttämään OpenFlow'ta "Southbound" API -rajapinnassa [76]. ONOS tarjoaa vision tulevaisuuden verkosta (kuva 29).



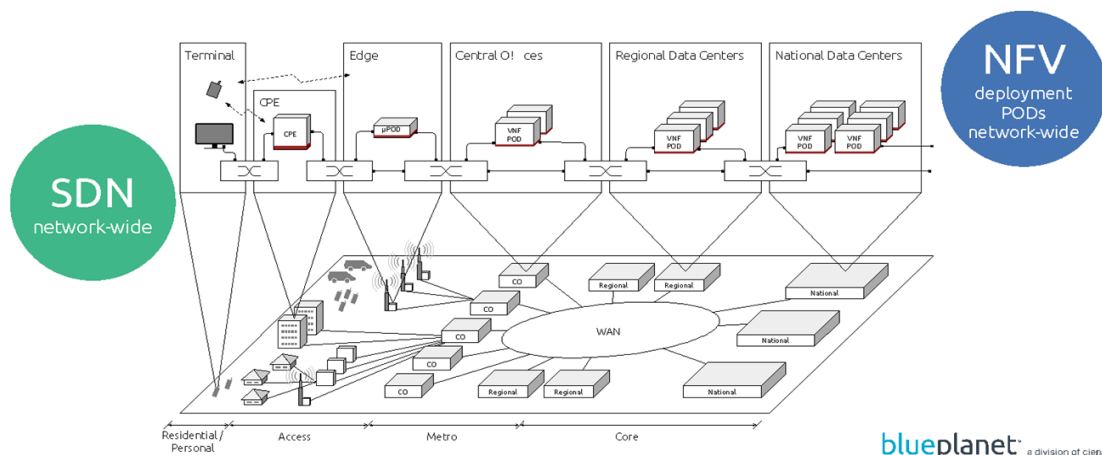
Kuva 29. Tulevaisuuden ONOS SDN -ohjelmistoarkkitehtuuri [75].

Arkkitehtuurin tasoja (kuva 29) ovat muun muassa Point Of Presence (POP), joka on liityntäpiste muualle internetiin, ja Central Office (CO), joka on puhelinkeskus. Nämä tasot rakennetaan kuten tietokonekeskus (Data Center). Tämä tarkoittaa, että ne rakennetaan Ethernet Switching Fabric -menetelmällä, jossa monta fyysistä kytkintä on yhdistynyt ja yhdessä laitteiston sekä ohjelmistokomponenttien kanssa ne muodostavat joustavan, automatisoidun kytkinverkon sekä käyttäytyy kuin, jolla on keskitty laitehallintaa. [77.] Runko (Core) ja suurkaupunkialueen optiset Metro Packet Optical -verkot rakennetaan edullisella laitteistolla, jossa kontrolleri ja kytkin on erotettu toisistaan.

ONOS SDN -verkonhallintaohjelma on asennettavissa pilvipalvelimelle virtuaalikoneeseen. Tuotekehitystarkoituksessa se on asennettavissa myös pieniin mikrokontrollereihin, kuten Raspberry Pi 3:een. [99.]

Kuvassa 30 on esitetty yhtenä esimerkkinä ONOS SDN -verkonhallinta-arkkitehtuurista tietoliikenne- eli telecom-laitevalmistaja Cienan käyttämä Blue Planet -arkkitehtuuri.

ONOS Use Cases in Service Provider Networks



Kuva 30. Ciena Blue Planet -arkkitehtuuri [76].

Blue Planet perustuu niin sanottu mikropalvelinarkkitehtuuriin (micro-services architecture), jossa käytetään kontainereita palvelujen orkestroinnissa laitteistoihin. Ohjelmistopalvelujen kehityksessä, testauksessa ja julkaisuissa käytetään DevOps-lähestymistapaa, joka mahdollistaa nopean ja sujuvan palvelujen julkaisun. Blue Planet mahdollistaa ohjelmistopalvelujen hallinnan aina loppukäyttäjien päätelaitteisiin saakka. [161.]

3.11 Avoimet SDN-kytkimet

Avoimia SDN-kytkimiä eli niin sanottuja White-Box-kytkimiä ovat muun muassa Broadcom, Marwell ja Intel/Fulcrum sekä tässä työssä käytetty kehityslaitteisto Raspberry Pi. Nämä laitteistot tarvitsevat kytkinohjelmiston toimiakseen.

3.11.1 White-Box-kytkinlaitteistoja

Avoimia SDN-kytkimiä kutsutaan White-Box-kytkimiksi. White-Box-kytkimellä tarkoitetaan kytkintä, joka ei ole välttämättä tunnetun valmistajan tuote ja joita valmistetaan suuria määriä [97]. Tällä tavoin laitteistojen hankintahinta on saatu alas. Kytkin suorittaa vain datan siirto-osuuden (Data Plane) eli pakettien kytkennän vuotaulukonsa pe-

rusteella. Kaksi yleistä White-Box-laitetoimittajaa ovat Broadcom ja Marwell. Muita on muun muassa Intel/Fulcrum. [4.]

White-Box- ja Black-Box-SDN-laitetoimittajia on listattu ONF:n, sdxcentralin ja Open-Flow'n wiki verkkosivuilla [85; 86; 78]. White-Boxina voi käyttää myös Raspberry Pi -kehityslaitteistoa, joka soveltuu hintansa, yleisyytensä ja Linux-pohjaisuuden puolesta hyvin muun muassa IoT-sovellusten alustaksi [99].

Avoimeen Raspberry Pi (R-Pi) -kehityslaitteistoon voidaan asentaa Mininet, joka sisältää Open vSwitch -ohjelmistokytkimen. Siihen voidaan asentaa myös kontrolleri-ohjelmisto. R-Pihin voidaan muodostaa Mininet-virtuaaliverkko, aivan kuten PC-ympäristöönkin. Mahdollisuutena on liittää R-Pi-kytkimeen toinen R-Pi tai muu vastaava laite ja käyttää sitä NFV "Middlebox"ina. Näin voidaan R-Pi kytkeä oikeaan verkkoon ja ohjata paketteja SDN-kontrollerilla ja käyttää tai kehittää NFV "Middlebox" -ohjelmistolaitteista palomuri (Firewall, FW), kuormanjako (Load Balancing, LB), valvonta (Monitoring) ja paketin syväluotaus (Deep Packet Inspection, DPI). [4.]

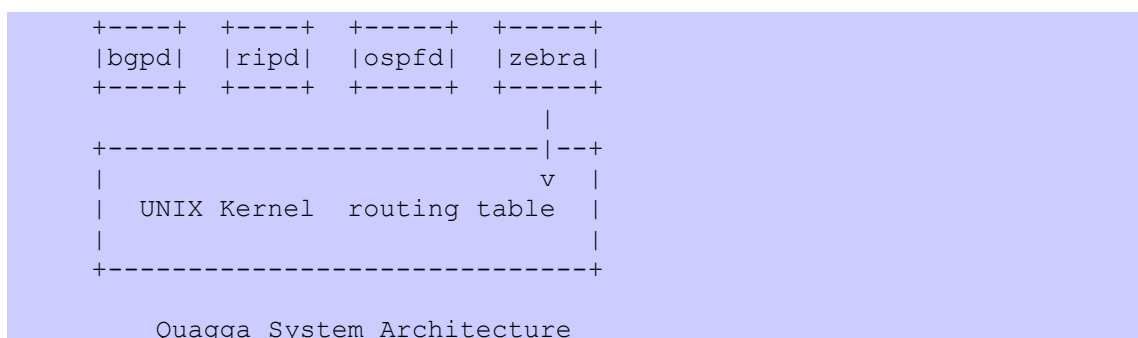
3.11.2 Open vSwitch -kytkinohjelmisto

Open vSwitchistä käytetään myös nimitystä (Open Virtual Switch, OVS). Open vSwitch -ohjelmisto on virtuaalinen ja monitasoinen verkkokytkin. Open vSwitch voi toimia kytkimenä pilvipalvelimien VM-koneissa ja myös laitepohjaisissa kytkimissä. [176.]

3.11.3 Quagga-kytkinohjelmisto

Tässä luvussa käsitellään Quagga-pohjaisen Vyatta-käyttöjärjestelmän hyödyntämistä SDN-reititintuotteiden rakentamiseksi.

Quagga-reititysohjelmisto on sisällytetty muun muassa Linux-ytimeen (kuva 31). Quagga sisältää yleiset reititysprotokollat, kuten Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS) ja Border Gateway Protocol (BGP). Myös Multiprotocol Label Switching (MPLS) -tuki on tulossa, ja se muun muassa mahdollistaa Asynchronous Transfer Mode (ATM)- ja Internet Protocol (IP) -yhteyksien toiminnan samassa kytkimessä. Quagga-ohjelmisto tarjoaa Cisco IOSin tapaisen komentotulkkirajapinnan (Command-line Interpreter, CLI) konfigurointikomentojen antamiseksi.



Kuva 31. Quaggan järjestelmäarkkitehtuuri [101].

Quagga-arkkitehtuuri mahdollistaa SDN-toiminnallisuuden eli reititysprotokollaohjelmiston siirtämisen pilvipalvelimeen.

SDN-reitittämiä valmistava yritys **Vyatta** hyödyntää tuotteissaan Linux Debianiin perustuvia ohjelmistoja, kuten Quagga-reititysohjelmistoa. Vyatta-tuoteperheet tarjoavat virtuaalireitittimen, joka perustuu Quaggaan, ja lisäksi virtuaalipalomuurin ja IPsec-perusteisen Virtual Private Network (VPN):n ja Secure Sockets Layer (SSL)-perusteisen OpenVPN:n. Vyatta tarjoaa tuen monille alustoille, kuten x86-laitteistoille ja VMware ESXi, Microsoft Hyper-V, Citrix XenServer, Open Source Xen ja KVM virtuaaliympäristöille. Vyatta on myyty Brocade Communicationille vuonna 2012. Vyattan avoin reititinohjelmistokehitys on nimeltään Vyatta Core Edition [100]. Brocade Communications järjestää muun muassa SDN/NFV-koulutusta [102]. Brocade tarjoaa ilmaista **Brocade Vyatta vRouter**- ja Brocade Vyatta Virtual ADX -tuotteilla tapahtuvaa 60 päivän NFV-koulutusta ja sertifiointia [103]. Myös Metropolia Ammattikorkeakoulussa on järjestetty vuonna 2014 viiden opintopisteen Brocade-kurssi ”Reititys Brocaden järjestelmillä”.

Yrityksoston jälkeen avoimen ohjelmiston Vyatta Core Editionin kehitystä jatkaa **VyOS**. VyOS on Linux-pohjainen reititysratkaisu, joka toimii Debian Linux -jakelun päällä ja on tällä hetkellä tuettuna x86- ja x86-64-laitteistoilla sekä VMware- ja Amazon EC2 -virtuaaliympäristöissä eli ei niin laajasti kuin Vyatta. VyOSin tavoitteena on jatkaa avointa ohjelmistokehitystä jatkaen siitä mihin Vyatta jäi. Viimeisin vakaa versio on VyOS 1.1.7 (Helium). Se tukee tällä hetkellä SDN-verkon reititystä, palomuuria ja VPN-toiminnallisuutta. Käyttöjärjestelmä on ladattavissa fyysiseen 32:n tai 64 bitin kytkimeen tai virtuaalikoneeseen. [104; 105].

3.12 Kaupalliset SDN-laitteistot

Yleisimpiä kaupallisia SDN-laitteistojen toimittajia ovat Big Switch Big Cloud Fabric, Plexxi Big Data Fabric, Brocade Vyatta Controller (esitelty aiemmin), HP Virtual Application Networks (VAN) SDN Controller/Virtual Cloud Networking (VCN), Juniper Contrail, Cisco Application Centric Infrastructure (ACI)/Application Policy Infrastructure Controller (APIC) ja lisäksi Alcatel-Lucent Nuage VSC (Nokia osti Alcatel-Lucentin) ja Cyan Blue Planet (Ciena on ostanut Cyanin) [85].

Suuria kaupallisia pilvipalveluiden tarjoajia ovat VMware ja Microsoft, jotka molemmat tarjoavat tuen SDN-kytkimille [107]. VMwaren tuote on VMware NSX (Nicira [108]) ja Microsoftin Windows Server 2012:n integroitu Hyper-V Extensible Virtual Switch.

SDN-kommunikointiprotokollaa käytetään yhdistämään ohjausosa (Control Plane) ja datan siirto-osa (Data Plane) toisiinsa. VMware NSX käyttää avointa OpenFlow'ta ja itse kehittämänsä Virtual Extensible Local Area Networkia (VXLAN). Microsoftin Hyper-V Extensible Virtual Switch pohjautuu OpenFlow-teknologiaan. Molempien valmistajien tuotteet ovat OpenFlow'n ansiosta liitettävissä muiden valmistajien SDN-tuotteisiin. VMware NSX- ja Microsoft Hyper-V -virtuaalikytkimistä kerrotaan lisää luvussa 3.14 "NV – verkon virtualisointi".

Kaupallinen ratkaisu tuotekehitysympäristön rakentamiseksi on käyttää esimerkiksi VMware-pilvipalvelua, Ciscon SDN-tietoliikenneverkkoa ja EMC:n tallennus- ja tietoturvaympäristöt yhdistävää Virtual Computing Environment (VCE) -ympäristöä, joka on vuonna 2014 julkaistu Suomessa [114]. Palvelu tarjoaa kokonaisratkaisun, johon kuuluu myös kattava koulutusjärjestelmä. Isojen toimijoiden valmiina kaupallisena ratkaisuna se on hinnoiteltu palvelun mukaisesti.

3.13 SDN/NFV-ohjelmoinnin haasteet

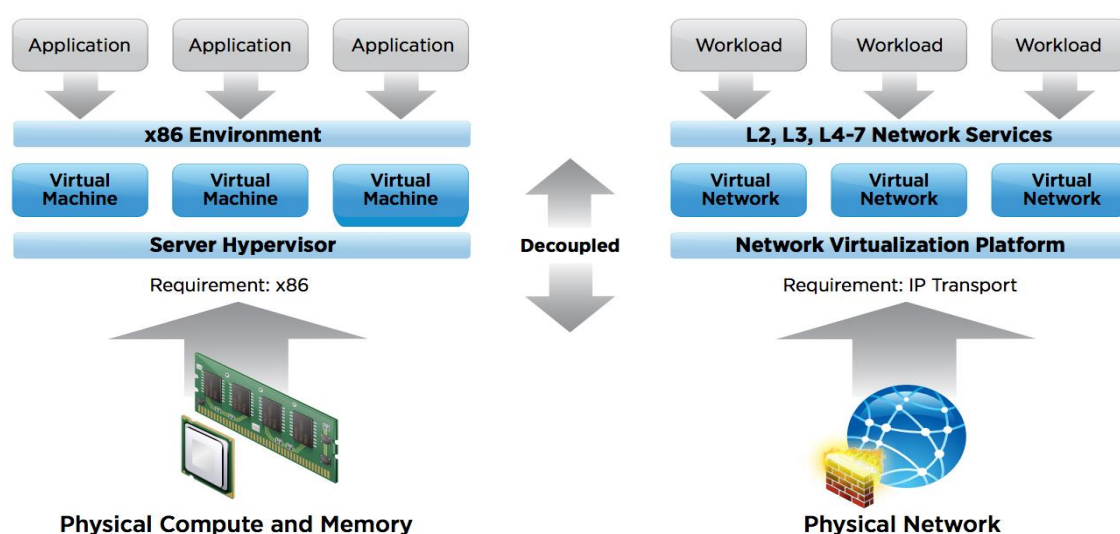
Esimerkiksi Princetonin ja Cornellin yliopistojen vetämän SDN-Freneticin kontrollerikieliprojektin ja sen jälkeisen SDN/NFV-kehitystyön haasteena on, kuinka SDN/NFV-infrastruktuuriin saadaan luotua osajayhteisö, joka kykenee kehittämään SDN/NFV-tuotteissa tarvittavat ohjelmistot ja osaa palvella näissä asioissa. Nykypäivänä oleellinen osa verkkojen ylläpitohenkilöstöstä ei kirjoita ohjelmistoja ja toisaalta ohjelmistoke-

hittäjistä suurin osa ei kovin paljoa tiedä tietoliikenneverkon toiminnasta. Toinen asia on, että tämän osaajayhteisön tulee kyetä tekemään paljon muutakin kuin vain kontrolloida verkkoa. Sen tulee kyetä integroimaan kokonaisuudet, kuten pilvipalvelimien palvelimet, joissa sovellutuksia suoritetaan, ja myös orkestroida nämä laajat palvelut, joissa tietoliikenneverkko on vain taustalla oleva yksi osa kokonaisuudesta. [6.]

3.14 NV – verkon virtualisointi

Network Virtualization (NV) eli tietoliikenneverkon virtualisoinnilla tarkoitetaan sitä, että fyysinen verkko, ohjelmistopohjainen verkko ja verkkotoiminnallisuus voidaan virtualisoida yhdeksi tai useammaksi loogiseksi virtuaaliverkoksi siten, että ne voivat toimia samanaikaisesti ja toisiaan häiritsemättä [106]. Tämän menetelmän etuna on, että nykyinen jo vuosikymmeniä kehitetty IP-pohjainen tietoliikenneverkko voidaan hyödyntää virtualisoimalla se ja siten päästään kehittämään uusia tietoliikenneverkkoarkkitehtuuria sen päällä, esimerkiksi SDN-teknologiaa, joka erottaa tietoliikennereitittimen ohjausosan (Control plane) datan siirto-osasta (Data plane). Lisäksi ei olla enää riippuvaisia verkkolaitetoimittajista kehitettäessä uutta tietoliikenneteknologiaa. [4.]

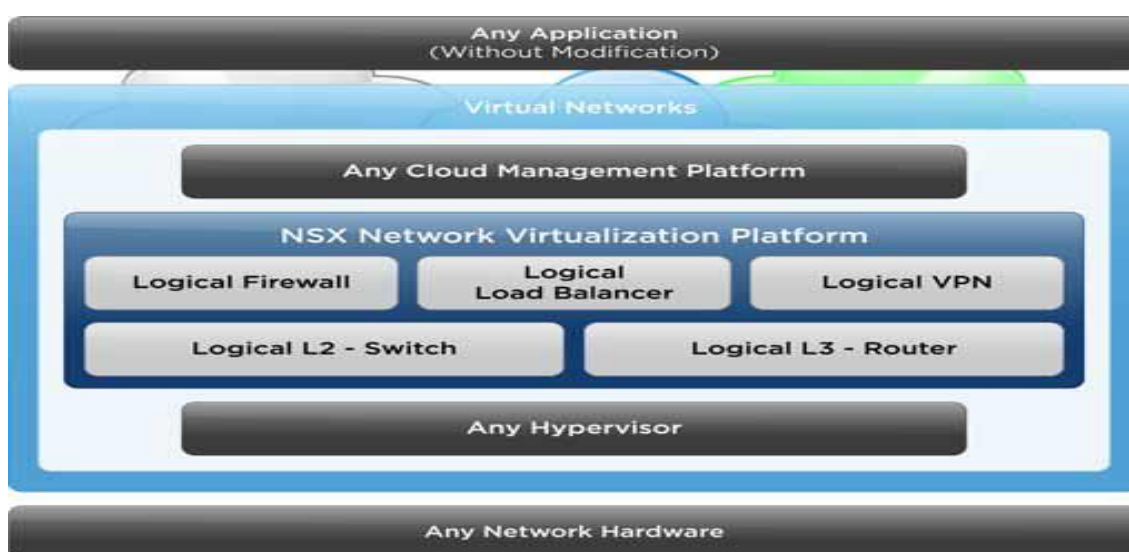
Kuvan 32 vasemmalla puolella on kuvattu pilvipalvelimesta tuttu ja aiemmin esitetty fyysisen PC:n prosessorin ja muistin virtualisointi hypervisorin avulla loogiseksi VM-koneen prosessoriksi ja muistiksi. Kuvan oikealla puolella on fyysisen verkon L2-, L3- ja L4–7-tasojen virtualisointi loogiseksi verkoiksi.



Kuva 32. Network Virtualization -arkkitehtuuri [108].

Aivan kuten hypervisoriarkkitehtuureja, myös verkon virtualisointiarkkitehtuureja on monenlaisia. Seuraavaksi kuvataan muutamia esimerkkejä kaupallisista verkon virtualisointiarkkitehtuureista.

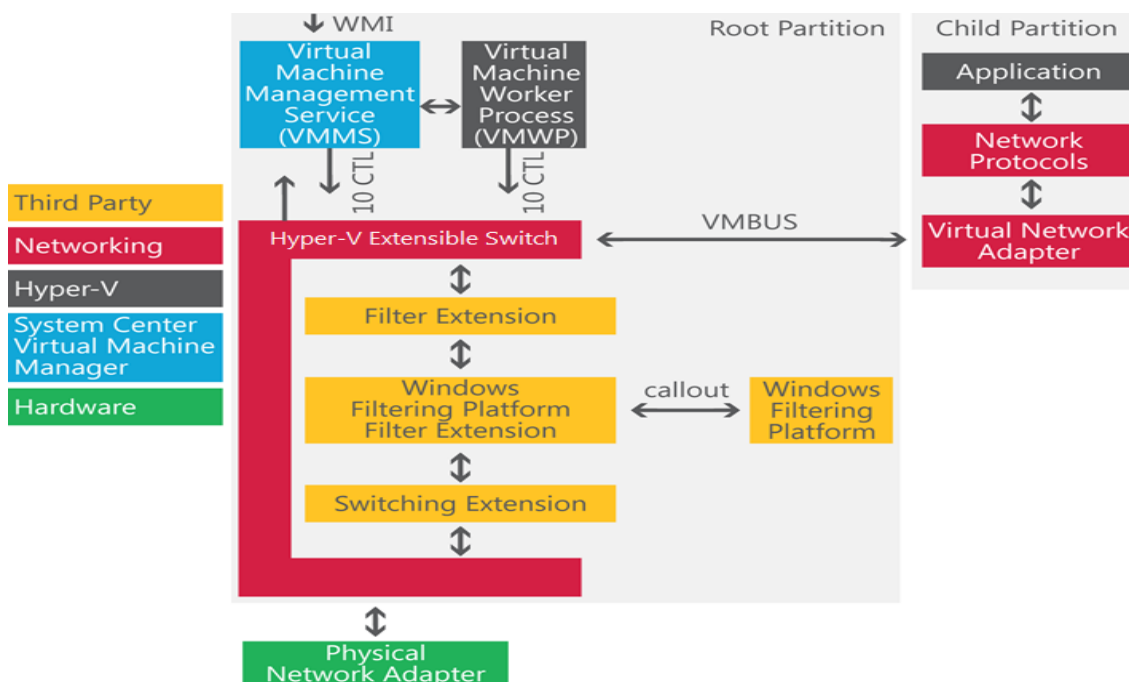
VMware osti vuonna 2012 SDN/NFV-startup-yrityksen Niciran, jonka teknologiaan VMware NSX perustuu. VMware NSX -virtuaalikytkin tukee erilaisia pilvipalvelin- ja -verkkokytkinarkkitehtuureja, kuten kuvan 33 VMware NSX -arkkitehtuurikuvasta ilmenee. [108; 112; 113].



Kuva 33. VMware NSX -arkkitehtuuri [108].

Kaupallisten verkkovirtualisointialustojen, kuten VMwareen ohjelmistotuotteilla rakennettujen verkkojen hinnoittelussa liikutaan kymmenien tuhansien eurojen luokassa. Prosessorikohtainen virtuaalikytkimen listahinta on noin 2 000–7 000 dollarin luokkaa riippuen muun muassa erilaisissa käyttökohteissa tarvittavista ominaisuuksista [109]. Ominaisuuksina ovat muun muassa tietoturvasäädö, mikrosegmentointituki ja kytkimen nopeus.

Microsoftin Windows Hyper-V Extensible Virtual Switch (kuva 34) on suunniteltu toimivaksi Microsoft-laiteympäristössä.



Kuva 34. MS Hyper-V Extensible Virtual Switch -arkkitehtuuri [111].

Microsoft Hyper-V Extensible Virtual Switchin tuki löytyy muun muassa Cisco Nexus 1000V -kytkimille.

3.15 NFV – verkkotoimintojen virtualisointi

Network Functions Virtualizationilla (NFV) eli verkkotoimintojen virtualisoinnilla tarkoitetaan tietoliikenneverkon yksittäisten fyysisten ja/tai ohjelmistojen sisältävien osakokonaisuuksien, kuten reititin, palomuurin ym., virtualisointia [115]. NFV on toinen osa SDN/NFV-verkkoympäristökokonaisuudessa. Telecom NFV -tuotekehityksessä voidaan virtualisoida tietoliikenneverkon laitepohjaiset NFV "middlebox"-laitteistojen toiminnot pilvipalvelimessa sijaitseviin ohjelmistopohjaisiin virtuaalikoneisiin tai kontainereihin. Nykyiset NFV "middlebox" -laitevalmistajat tutkivat mahdollisuuksia lisensoida laitepohjaisten laitteistojen ohjelmistot, jotta ne voisivat myydä tuotteensa ohjelmistona teleoperaattoreille. SDN:n tehtävänä on ohjata tietyn verkon liikenne ohjelmistopohjaisiin NFV "middlebox" -toimintoihin. [4.]

Näitä **NFV "middlebox" -toimintoja** ovat muun muassa palomuri (Firewall, FW), yhdyskäytävä (gateway, GW), kuormanjako (Load Balancing, LB), valvonta (Monitoring), paketin syväluotaus (Deep Packet Inspection, DPI), matkapuhelinverkko (mobile

network), yritysten palvelut ja sisällön jakopalvelut (Content Distribution Network, CDN), web-palvelin, sovellus ja tietokanta (database, DB). [4.]

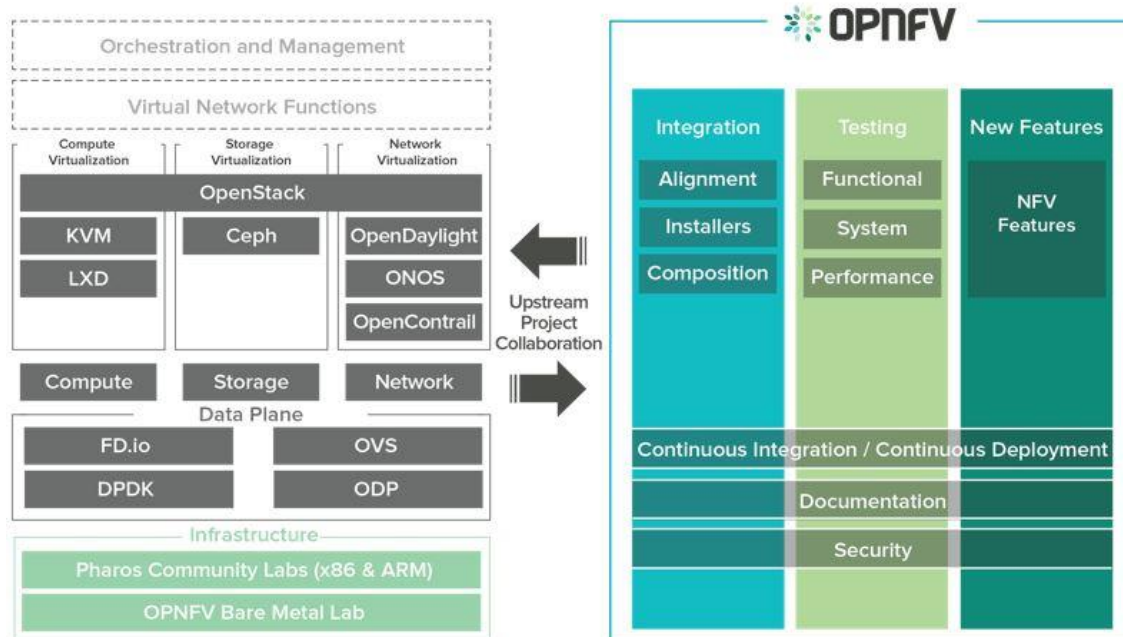
3.16 OPNFV – avoin NFV-yhteisö

Avoin NFV-yhteisö (Open Platform for NFV, OPNFV) ja sen samanniminen "carrier-grade"-tason integroitu ja avoin ohjelmistoalusta OPNFV nopeuttaa uusien NFV-tuotteiden ja -palvelujen julkaisua. OPNFV:n kehittämisessä on mukana laaja joukko yrityksiä, kuten AT&T, Brocade, China Mobile, Cisco, DELL, EMC, Ericsson, HP, Huawei, IBM, Intel, Juniper, Nec, Nokia, Docomo, RedHat, Telecom Italia, Vodafone, ZTE ja VMware. [116].

Yhteisö on julkaissut ensimmäisen OPNFV-ohjelmistoalustaversion Arnon 4.6.2015. Toinen versio julkaistiin 1.3.2016 ja on nimeltään Brahmaputra 1.0n. Ohjelmistot sisältävät NFV Infrastructure (NFVI)- ja Virtual Infrastructure Manager (VIM) -komponentit, jotka perustuvat ETSI NFV -arkkitehtuuriin. Ohjelmistoalusta sisältää jatkuvan integroinnin (continuous integration), automaattisen julkaisun ja testauksen ohjelmistoprojekteilte, kuten Ceph (storage ajurit), KVM, OpenDaylight, OpenStack ja Open vSwitchiin (OVS). [117].

Brahmaputra-versioon on tullut lisäominaisuuksia, kuten ONOS- ja OpenContrail SDN-ohjausosan tuki ja datan siirto-osaan Data Plane Development Kit (DPDK)- ja Open Distributed Processing (ODP) -tuki sekä IPv6-, Service Function Chain (SFC)- ja L3VPN-tuki. [117].

Kolmas ja tällä hetkellä viimeisin versio on Colorado (kuva 35).



Kuva 35. OPNFV-arkkitehtuuri [117].

Colorado-versioon on lisätty ja paranneltu ominaisuuksia, kuten SDN L2 ja L3 VPN sisältäen Quagga BGP -reitityksen, moniarkkitehtuuriset laitteistot x86 ja ARM, täysin integroitu testaus ja Continuous Integration (CI)/Continuous Delivery (CD) DevOps-prosessit ja Linux Container Daemon (LXD). LXD:llä on luotu parempi käyttöliittymä palveluprosessin (daemon) avulla Linux Container (LXC):n päälle ja yhteensopivuus OpenStack Novaan. Lisäksi on tuki yhteisöjen kautta, kuten FD.io (Fast data - Input/Output), joka on Software Defined Infrastructure (SDI) -kehittäjille suunnattu ja sisältää monia kehitysprojekteja. [117.]

Mainitut lisäykset mahdollistavat kehittäjien ja käyttäjien automatisoidusti asentaa ja laajentaa OPNFV-alustaa. Alusta on tarkoitettu kaikille, jotka tutkivat NFV:n käyttöönottoa, kehittävät NFV-sovelluksia tai ovat kiinnostuneita NFV:n suorituskyvyn ja käytön testauksesta. [117.]

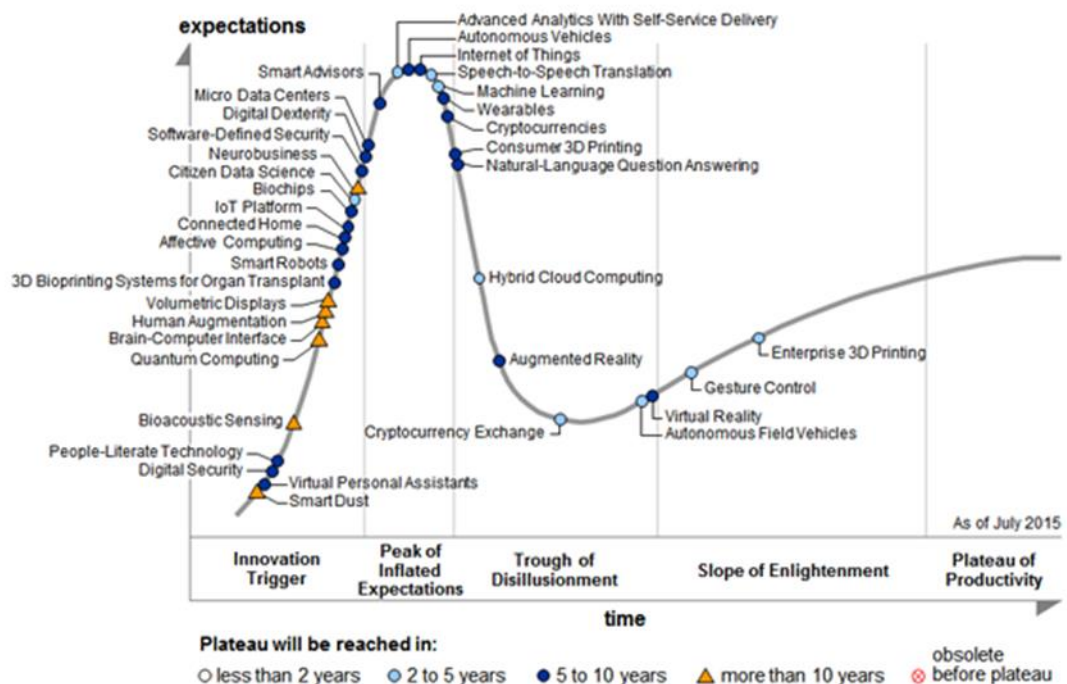
Tämän insinööriyön puitteissa OPNFV-yhteisöön ja sen Pharos-projektin testilaboratorioon tutustuttiin teoriatasolla. Testilaboratorio on yhteisöön kuuluvien yritysten käytössä. Käyttäjinä ovat muun muassa verkkolaittevalmistajat Ericsson ja Nokia. [117.]

Esineiden internetin arvioidaan tuovan Suomeen noin 50 000 uutta työpaikkaa vuoteen 2023 mennessä ja mullistavan maailmaa yhtä paljon kuin sähkön ja tietokoneiden tulo [121].

Esineiden internet on välivaihe matkalla seuraavaan teknologiseen vaiheeseen eli ohjelmoitavaan maailmaan (programmable world), jonka SDx-ohjelmistoalustat mahdollistavat. Tällä tarkoitetaan esimerkiksi itseohjautuvia autoja, joiden ohjauksesta huolehtivat sensori- ja tietoliikenneverkot.

4.2 IoT:n yleistyminen

Kuvassa 37 on Gartnerin ”Hype Cycle for Emerging Technologies” -analyysi heinäkuulta 2015. Se kuvaa nousevien teknologioiden yleistymistä ja mahdollista yleistä käyttöönottoa. [122.]

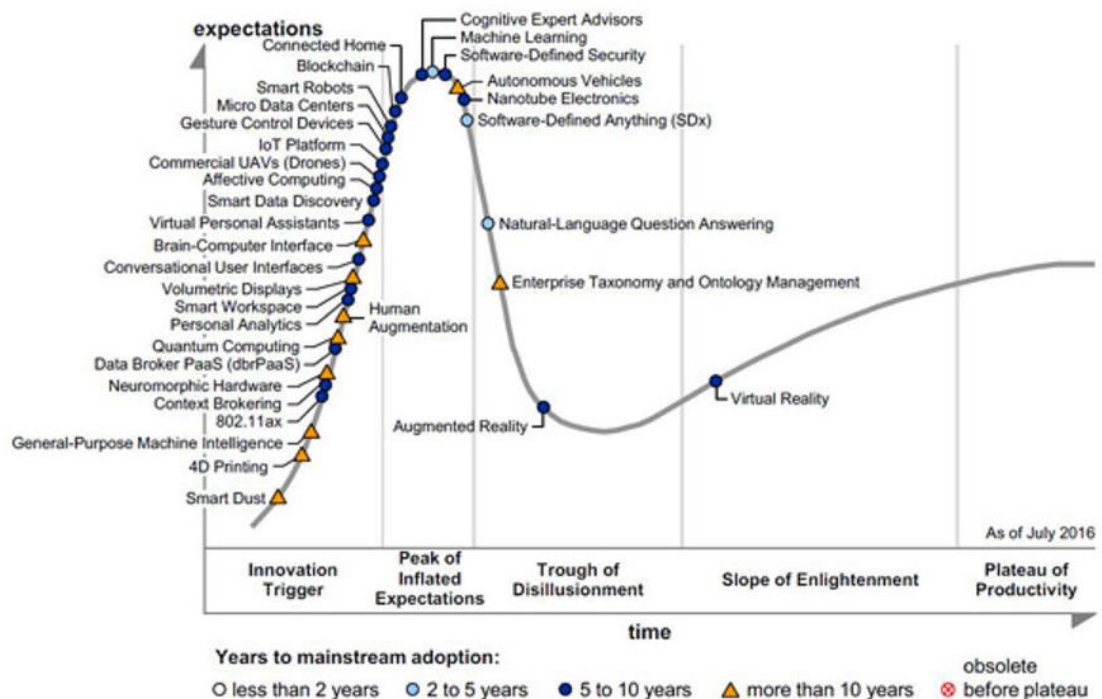


Kuva 37. Gartnerin ”Hype Cycle for Emerging Technologies, 2015” -analyysi nousevien teknologioiden yleistymisestä [122].

Kuvasta 37 voidaan nähdä Internet of Things (IoT) -teknologian yleistyminen. IoT:n osalta ”markkinointihypetys” on huipussaan, ja sen on arvioitu olevan 5–10 vuoden

kuluttua käyrän tasaisella, toteutusvaiheen puolella. Tällöin IoT-tekniologia olisi tuottavimmillaan. Samaa 5–10 vuoden päässä olevaa tuottavuushuippua Gartner on arvioinut myös SDN:lle, kuten aiemmin on esitetty. SDN on kriittinen tekijä IoT-palvelujen mahdollistajana. [82.]

Kuvassa 38 on Gartnerin vuoden 2016 ”Hype Cycle for Emerging Technologies” -analyysi. Merkillä pantavaa siinä on, että SDN-, NFV- ja IoT-tekniologiat on yhdistetty käsitteeksi ”Software-Defined Anything” (SDx).



Kuva 38. Gartnerin ”Hype Cycle for Emerging Technologies, 2016” -analyysi nousevien teknologioiden yleistymisestä [123].

Digitalisaation mahdollistavat teknologiat (kuva 38) ovat kaikki niin sanotusti ”digitaalisia”. Uusien teknologioiden nimitykset voivat vaihdella riippuen siitä, mikä taho ne esittää. Esimerkiksi SDxCentral käyttää SDx-teknologiasta nimitystä Software-Defined Everything [1]. SDx-teknologian on kuvattu saavuttavan käyrän tasaisen vaiheen jo 2–5 vuoden kuluttua ja useiden muiden digitalisaatioiden teknologioiden 5–10 vuoden kuluttua.

Pilvipalvelintekniologiaa kehittävän yrityksen VMwaren tutkimuksen mukaan internetiin kytkettyjen IoT-laitteiden määrä kasvaa taulukon 2 mukaisesti [110].

Taulukko 2. Tutkimus VMwaren vuonna 2016. PC-, tabletti- ja mobiililaitteiden lukumäärien ja IoT-laitteiden lukumäärien vertaus vuosina 2006, 2011, 2016 ja 2021. [110.]

<i>IT-palvelut</i>	<i>2006</i>	<i>2011</i>	<i>2016</i>	<i>2021</i>
<i>PC-, tablet- ja mobiililaitteet (miljardia kpl, eli 10⁹)</i>	2,9	5,7	8,2	8,7
<i>IoT-laitteet (miljardia kpl, eli 10⁹)</i>	-	1,2	4,1	18,0

WMwaren tutkimuksen mukaan internetiin kytkettyjen PC-, tabletti- ja mobiililaitteiden lukumäärä ei enää juurikaan kasva. Sen sijaan IoT-laitteiden lukumäärä kasvaa 4,5-kertaisesti seuraavan viiden vuoden aikana. [110.] Vastaavan Gartnerin ennusteen mukaan vuonna 2020 on 20,8 miljardia laitetta [124]. Vertailun vuoksi maailman väkiluku oli vuonna 2016 seitsemän miljardia [125].

4.3 IoT-sovelluskohteita

IoT tarjoaa lupaavia mahdollisuuksia yrityksille lähinnä toimintojen tehostamiseen ja kilpailukyvyyn parantamiseen sekä uusien palvelujen tuottamiseen. SDN on kriittinen tekijä IoT-palvelujen mahdollistajana [82].

IoT:llä saadaan esimerkiksi liikenne sujuvaksi ja vähemmän ympäristöä kuormittavaksi, kun liikennettä hidastavista liikennevaloista voidaan siirtyä joustavaan liikenteenohjaukseen. Vastaavia ohjelmoitavan maailman sovelluskohteita löytyy ympäristömme muista vastaavista verkoista, kuten yhteiskunnan sähkö-, vesi-, viemäri- ja kaukolämpöverkoista. Sensoriverkkojen automaattisen valvonnan avulla voidaan poistaa verkko-vootojen aiheuttamat hävikit.

IoT:n ja ohjelmoitavan maailman ohjelmistoissa käytetään sulautettuja ohjelmistoja laiteläheisissä kohteissa. Varsinainen tiedon hallinta eli sen käsittely ja tallennus tapahtuvat tietoliikenneverkon reunoilla niin sanotuissa sumuverkoissa (Fog Networks) ja pilvipalvelinympäristössä. Vallitsevina ajatuksina ovat avoin ohjelmistokehitys pohjautuen Linux-käyttöjärjestelmään ja sen eri kehityshaaroihin. [4.]

4.4 IoT-alan toimijoita Suomessa

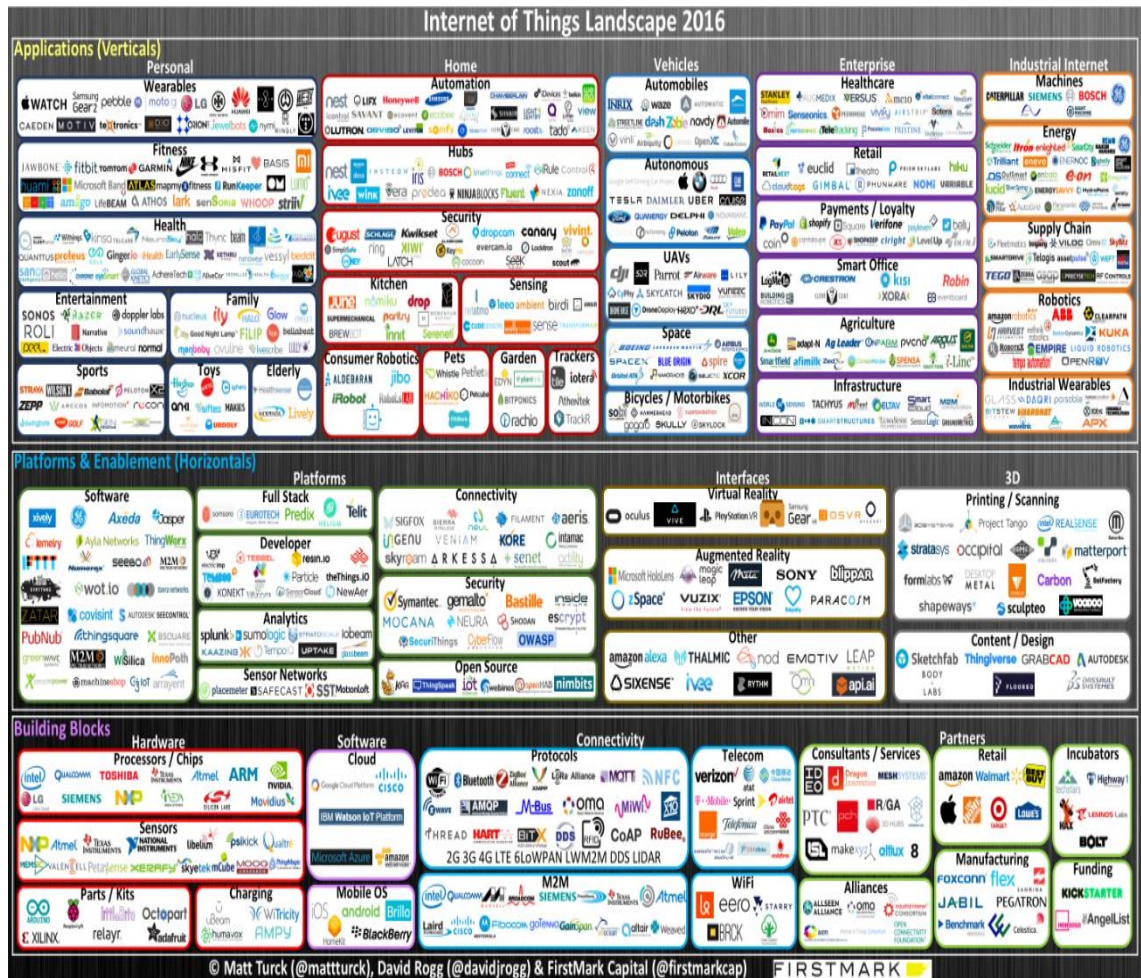
Suomessa on monia IoT-alan toimijoita, jotka pyrkivät hahmottamaan IoT:tä kokonaisuutena ja edistämään sen käyttöönottoa. Yksi jo päättynyt kansallinen IoT-projekti oli DIGILE. Projektiin osallistui yli 350 tutkijaa ja kansainvälistä asiantuntijaa. [128]. Toinen on suomalainen IoT-yhteisö Finnish Industrial Internet Forum (FIIF), jonka jäsenenä on muun muassa yrityksiä, yliopistoja, korkeakouluja ja rahoittajia [129].

Yksi suurimmista sijoittajista on Nokia Growth Partners, jonka mukaan erityisen kiinnostavia IoT-startup-hankinnat ovat IoT-ohjelmistopino ja verkkopuolella, missä on paljon mahdollisuuksia, kun taas laitteistopuolella niitä on vähemmän [130]. IoT-ohjelmistopino ja verkkopuolen mahdollisuuksia tukee Gartnerin ennustus vuodelle 2020, jolloin yli puolet kaikista uusista PaaS-alustoille kehitetyistä sovellutuksista tulisi olemaan IoT-keskeisiä sovellutuksia [131].

4.5 IoT-maisemakuva

IoT:stä odotetaan suurta tulonlähdettä alalla toimiville yrityksille, ja siten jokainen pyrkii ottamaan mahdollisimman suuren osan tulevasta yhteisestä ”kakusta”. Yhtenä tapana vallata markkinoita on luoda oma IoT-kehitysalusta eli niin sanottu ”IoT framework”. Niitä on jo nyt lukuisia, ja lähes jokaisella isolla yrityksellä alkaa olla omansa, kuten esimerkiksi siruvalmistajilla (Intel, Qualcomm, ARM), teleoperaattoreilla (AT&T), pilvipalvelinalustoilla (IBM, Cisco), mobiilialustoilla (Android, Apple ja Microsoft) ja työkalu ympäristöjen kehittäjillä (Eclipse). Erikseen on vielä mainittava Google ja Amazon niiden suuren markkinapotentiaalin vuoksi.

Kuva 39 havainnollistaa maailmanlaajuisesti IoT-alalla toimivien yritysten sijoittumista IoT-maisemakuvassa. Maisemakuvan on laatinut yhdysvaltalainen FirstMark-riskipääomasijoittaja, jolla on sijoittajan näkökulma asiaan.



Kuva 39. IoT-ympäristön maisema [132].

Kuvasta saa yleisnäkemyksen IoT:n käyttökohteista, ohjelmistoalustoista ja laitteistoista sekä protokollista.

4.6 IoT-ekosysteemi

IoT-ekosysteemi koostuu markkinalähtöisistä erilaisiin ympäristöihin suunnitelluista alustoista. Näitä on muun muassa sähköverkko- (Smart Grid), koti- (Connected home), terveydenhoito- (Smart Health) ja kaupunkiympäristö (Smart Cities). Kuvassa 40 on vain yksi yleisluonteinen esimerkki siitä kuinka IoT-ekosysteemiä ja sen tasoja voidaan kuvata. Kaikissa erilaisissa IoT-ekosysteemikuvauksissa on kuitenkin vastaavanlaisia tasoja. IoT-ekosysteemeissä monet yritykset ja muut yhteistyötahot valmistavat näihin ympäristöihin yhteensopivia laitteistoja ja ohjelmistoja.



Kuva 40. IoT-ekosysteemi [133].

Kuvassa 40 esitetyn mukaisesti sensoritietoa hankitaan esimerkiksi Smart Cities -ympäristöstä L1-fyysisen tason sensoreilta ja yhteenliitetään L2-siirtokerrostason ja L3-verkkokerrostasojen kautta eteenpäin ja integroidaan yhteen pilvipalvelimessa sijaitsevalle tietokannalle. Tästä tiedosta jalostetaan analytiikan ja sovellusohjelmien avulla loppukäyttäjää hyödyttäviä palveluja.

4.7 IoT-protokollia ja standardeja

IoT-protokollia ja standardeja on lukuisa määrä. Kuvassa 41 niistä on kuvattu vain osa. Kattavamman kuvan saa tutustumalla standardoimisjärjestöihin ja niiden verkkosivuihin. Standardoimisjärjestöjä esitellään tämän dokumentin luvussa 4.9.1 "Euroopan IoT-standardointi".

Session		MQTT, SMQTT, CoRE, DDS, AMQP, XMPP, CoAP, ...	Security TCG, Oath 2.0, SMACK, SASL, ISASecure, ace, DTLS, Dice, ...	Management IEEE 1905, IEEE 1451, ...
Network	Encapsulation	6LoWPAN, 6TiSCH, 6Lo, Thread, ...		
	Routing	RPL, CORPL, CARP, ...		
Datalink		WiFi, Bluetooth Low Energy, Z-Wave, ZigBee Smart, DECT/ULE, 3G/LTE, NFC, Weightless, HomePlug GP, 802.11ah, 802.15.4e, G.9959, WirelessHART, DASH7, ANT+, LTE-A, LoRaWAN, ...		

Kuva 41. IoT-protokollatasot ja -standardit [133].

Sensorit muodostavat L1-tason fyysisen kerroksen, jonka päällä ovat L2-siirtokerros (Datalink), L3-verkkokerros (Network) ja L5-istuntokerros (Session).

Tärkeimpiä L2-tason standardeja ovat yleisesti käytetyt Wireless LAN (WLAN)-, Bluetooth-, ZigBee-, Z-wave- ja Long Range Wide Area Network (LoRaWAN)- sekä Wirepas- ja SIGFOX-sensoriverkkoteknologiat, joita tutkitaan lähemmin luvuissa 4.7.1 - 4.7.6. Kaikilla näillä tekniikoilla voidaan muodostaa myös IoT-sensoriverkko.

4.7.1 WLAN-standardi

Wireless LAN (WLAN) on langaton lähiverkkoteknologia, jolla laitteistot voidaan yhdistää langattomasti toisiinsa. Siitä käytetään myös nimitystä WiFi. Se noudattaa standardia IEEE 802.11. Yleisin käytössä oleva versio on 802.11g, jonka maksimi tiedonsiirtonopeus on 54 Mbit/s [134], joka on riittävä nopeus IoT-sovellutuksiin. Paristokäyttöisissä IoT-laitteissa virrankulutus tulee tällä tekniikalla ongelmaksi. WLAN-tekniikkaan perustuvia IoT-laitteita käytetään kohteissa, joissa tarvitaan suurta tiedonsiirtonopeutta.

4.7.2 Bluetooth-standardi

Bluetooth on langaton verkkoteknologia (Wireless Personal Area Network, WPAN), kuten myös ZigBee. Tavallisessa Bluetooth 3.0 -versiossa aktiivisia asiakaslaitteita voi olla seitsemän. [137.]

Nykyään yleisesti käytössä olevasta Bluetooth 4.0 -versiosta käytetään nimityksiä Bluetooth LE, BLE ja Bluetooth Smart. Siinä aktiivisten asiakaslaitteiden määrä riippuu toteutuksesta. Bluetooth 4.0 -teknologia on hyödyllinen, kun halutaan siirtää esimerkiksi ZigBee-sensoriverkosta tietoa matkapuhelimen kautta internetiin ja sieltä pilvipalvelimelle. Uusimmat matkapuhelimet tukevat sitä ja mahdollistavat siten matkapuhelimen toimimisen isäntä-laitteena. [138.]

Uusin versio Bluetooth 5 on julkaistu vuonna 2016. Se mahdollistaa Bluetooth-tekniikan käytön myös IoT-sensoriverkoissa. [139.]

4.7.3 ZigBee-standardi

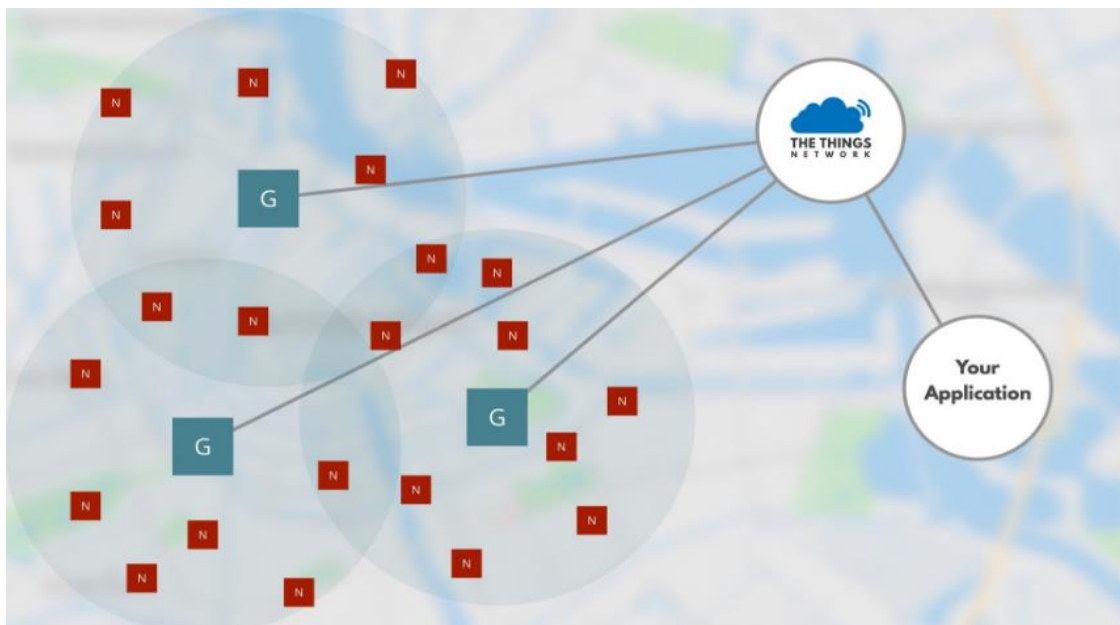
ZigBee on lyhyen kantaman WPAN-verkko, jossa solmukohdan muodostavat ZigBee-laitteet. Solmujen välimatka toisiinsa on kymmeniä metrejä ja pisimmillään noin 100 metriä. Solmuja voi teoriassa olla 65 536 laitetta. [135; 136.]

ZigBee perustuu IEE 802.15.4 -standardiin, joka määrittelee OSI-tasot, L1- eli fyysisen tason ja L2- eli siirtoyhteystason [136]. Kodin IoT-laitteissa on yleistynyt Z-Wave-teknologia.

4.7.4 LoRaWAN-standardi

Long Range Wide Area Network (LoRaWAN) on pitkän matkan Media Access Control (MAC) -protokolla. Pienitehonkulutuksinen verkko Low Power Wide Area Network (LPWAN) perustuu Long Range (LoRa) -radiomodulaatiotekniikkaan. OSI-mallissa LoRa on L1-tasolla ja LoRaWAN on L2- ja L3-tasoilla. LoRaWAN-teknikkaa on tarkoitettu käytettäväksi langattomissa paristokäyttöisissä IoT-laitteissa.

Kuvan 42 LoRaWAN-verkko käyttää "star-of-stars" -topologiaa, jossa jokainen yhdyskäytävä (gateway, G tai GW) välittää viestejä solmujen (Node, N) välillä. Solmut käyttävät yhden hypyn (single hops) langatonta yhteyttä yhdelle tai usealle yhdyskäytävälle. [140.]



Kuva 42. LoRaWAN-standardin mukainen topologia [140].

LoRaWAN-tekniikan käyttämät taajuuudet vaihtelevat eri maanosissa, ja säätelystä vastaavat radioverkkoviranomaiset. Tiedonsiirtonopeudet ja -kesto vaihtelevat luokituksen Class A, Class B ja Class C mukaan. Class A on tarkoitettu IoT-laitteistolle, joka käyttää vain lyhyitä ja vähän tiedonsiirtokaistaa vaativia viestejä. Class B on tarkoitettu kaksisuuntaiseen kommunikointiin, ja Class C on tarkoitettu lähes jatkuvaan tiedon välitykseen. LoRaWAN-standardin kehitystä ohjaa LoRaWAN-allianssi. [141.]

4.7.5 Wirepas-sensoriverkko

Wirepas-sensoriverkko on IoT-verkko, joka ei tarvitse Radio Access Network (RAN) -verkkoinfrastruktuuria toimiakseen. Siinä IoT-sensorilaitteet eli solmut toimivat myös reitittiminä ja muodostavat itsenäisesti sensoriverkon ilman keskitettyä verkonhallintaa, konfigurointia tai laitteistoja. Se vaatii toimiakseen yksinkertaisen mikrosirun ja Wirepas Connectivity -ohjelmiston, jotka voi periaatteessa liittää mihin tahansa laitteeseen. [142.]

4.7.6 SIGFOX-sensoriverkko

SIGFOX on globaali ja langaton IoT-sensoriverkko, joka on ranskalaisen yrityksen suunnittelema ja hallinnoima. Se on täysin riippumaton yleisistä verkoista, kuten tele-

kommunikaatioverkoista. Se on suunniteltu hyvin pienien datamäärien välittämiseen ja hyvin pienin kustannuksin, edullisimmillaan noin 1 euro/laite/vuosi. SIGFOX IoT-verkko on laajentunut viime vuosina huomattavasti muun muassa Suomeen ja on nykyään yksi varteenotettavimmista IoT-sensoriverkoista. [143.]

4.7.7 4G- ja 5G-standardit

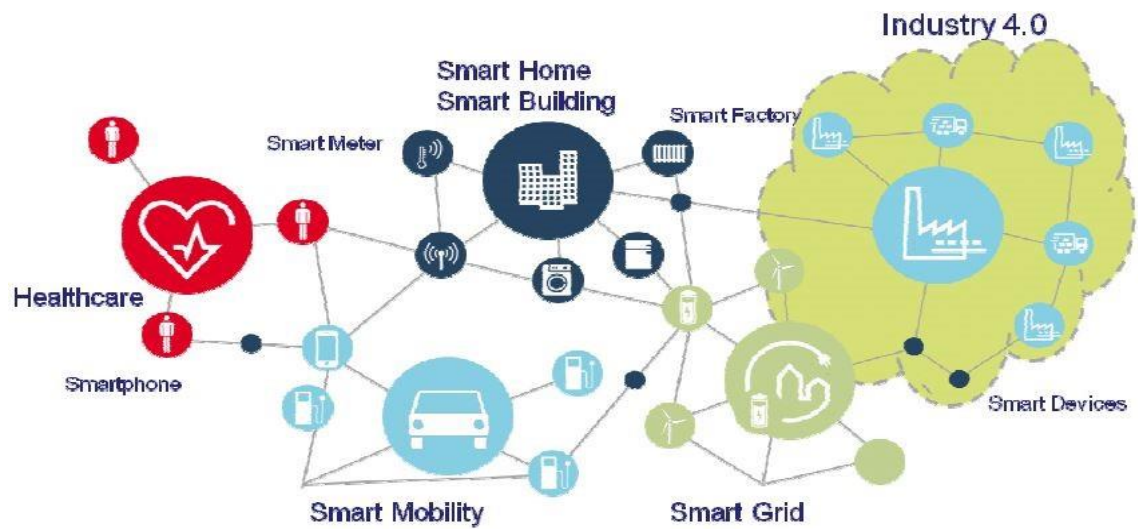
Nykyinen 4G-verkko sisältää lähinnä datansiirron [126]. Tulevaan SDx:ään eli ohjelmoitavaan maailmaan kuuluva 5G-verkkoteknologia sisältää myös palvelut [144]. Käytännössä 5G-verkko rakentuu nykyisen 4G-verkon rinnalle. 5G-standardoinnin ensimmäinen versio alkaa valmistua vuonna 2017 ja testiverkoja rakennetaan mahdollisesti muun muassa Tokion olympialaisiin, jotka ovat vuonna 2020. Puhelimia ja muita laitteita alkanee ilmestyä vuonna 2020. [144.]

5G-standardin suunnittelu on vielä kesken. Keskeiset määriteltävät asiat ovat verkon nopeus, esineiden internet ja kriittisen kommunikaation varmistaminen, joiden painotuksista käydään keskustelua. 5G-standardin oletetaan valmistuvan vuoteen 2020 mennessä. Tulevat tietoliikenteen verkkotekniikat ovat ohjelmistopohjaisia, ja laitteistojen osuus on niissä vähäinen. [127]

5G-verkon tavoitteena on päästä alle 1 ms:n viiveeseen tiedonsiirrossa lähettäjälaitteelta tukiaseman kautta vastaanottajalaitteelle. Nykyisellä 4G-verkolla viive on pienimmillään 10 ms, mutta yleensä se on 13–20 ms. Tällä on suuri merkitys esimerkiksi älyliikenteen toiminnan kannalta, jossa viiveiden tulee olla mahdollisimman pieniä, jotta ehditään reagoida liikenteessä eteen tuleviin tilanteisiin. Tämä voidaan saavuttaa siten, että IoT-ohjelmistoja suorittavat laitteistot ovat lähellä IoT-sensoreita ja ohjattavaa kohdetta. [144.]

4.8 Teollinen internet 4.0

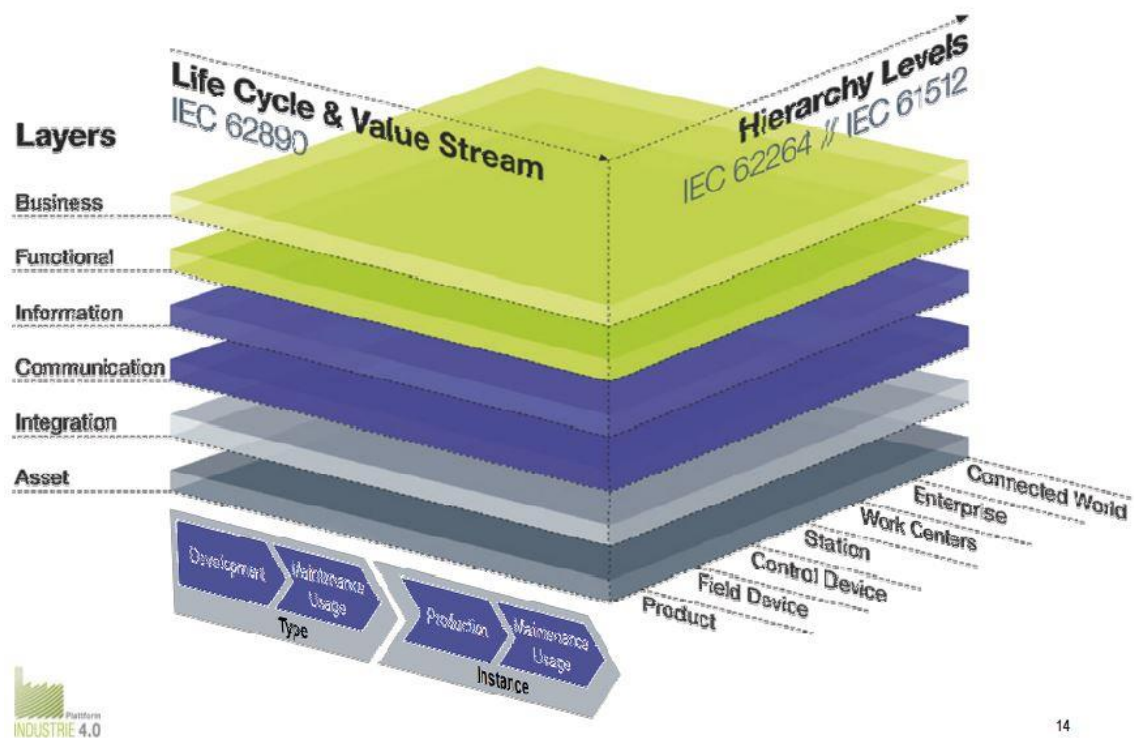
Teollinen internet on yksi osa-alue esineiden internetistä, kuten aiemmista esineiden internetosion kuvista ja seuraavasta kuvasta 43 voidaan havaita. Kuva havainnollistaa, kuinka esineiden internet ja sen palvelut sekä Saksasta lähtöisin oleva teollinen internet 4.0 (Industry 4.0) liittyvät toisiinsa.



Kuva 43. Esineiden internet ja palvelut sekä teollinen internet 4.0 [145].

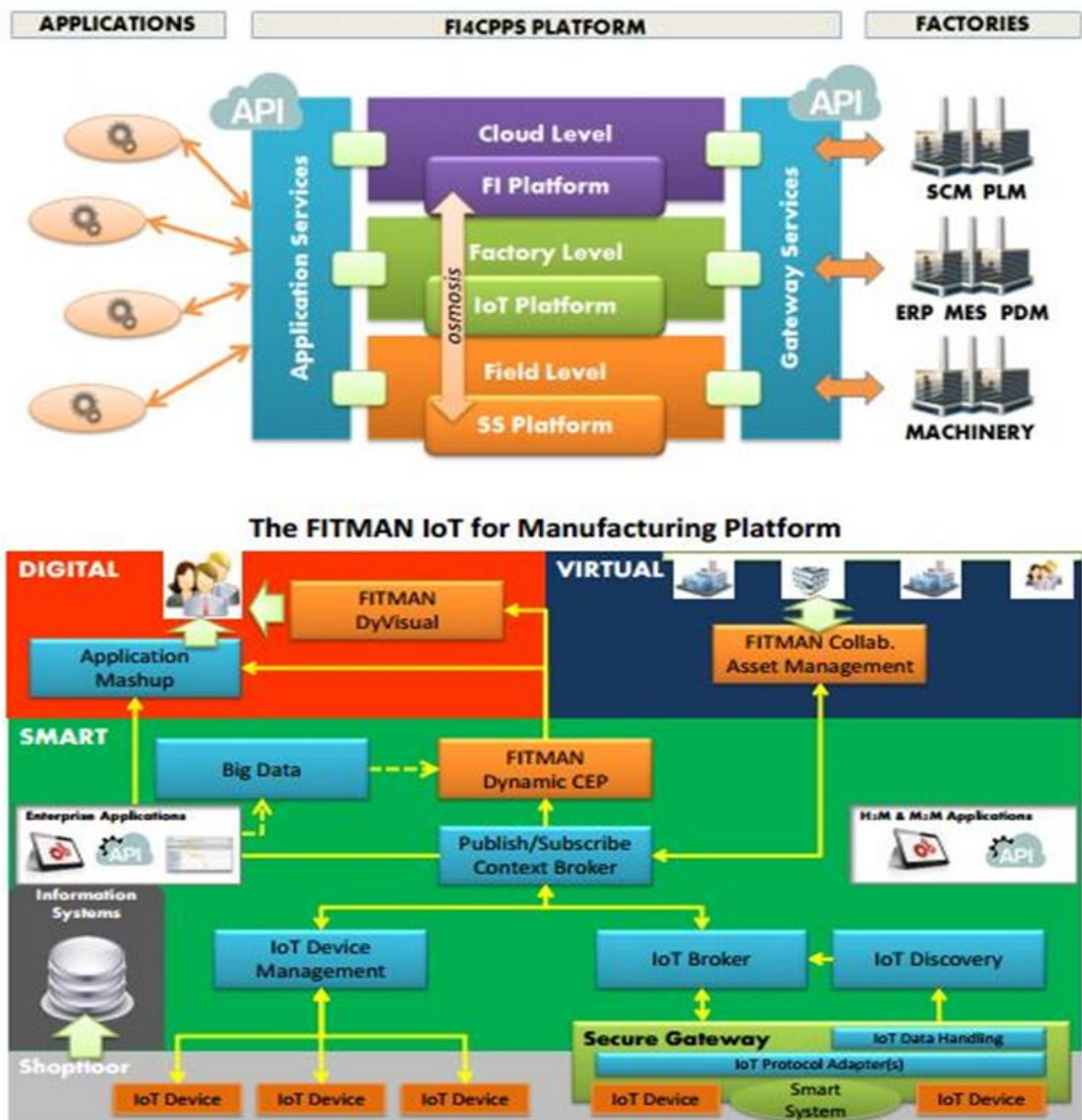
Teollinen internet 4.0:n standardissa (Reference Architecture Model Industry 4.0 (RAMI4.0) (kuva 44) ja Euroopan unionin osarahoittamassa FIWARE-alustassa (kuva 46) on arkkitehtuurisesta näkökulmasta katsottuna paljon yhtäläisyyksiä.

Reference Architecture Model Industry 4.0 (RAMI4.0)



Kuva 44. Teollinen internet 4.0 (RAMI 4.0) [145].

Johtopäätelmänä tästä on, että teollinen internet 4.0:n ja FIWARE-alustan voisi yhdistää ja kehittää FIWARE-alustaa hyödyntävä teollisen internetin alusta. Tässä asiassa onkin jo EU-tasolla edetty [147]. Euroopan unioni järjestää työryhmiä, kuten "Platforms for connected Factories of the Future", joissa määritellään tulevaisuuden yhteisiä standardeja eurooppalaiselle teollisuudelle. Kuvassa 45 on teollisuuden FITMAN IoT -alusta.



Kuva 45. Teollisuuden FITMAN IoT -alustan arkkitehtuuri [146].

Teollisuuden FITMAN IoT -alusta pohjautuu FIWARE-alustaan. Pyrkimyksenä on luoda Euroopan unionin alueelle yhtenäinen teollisen internetin alusta. FITMAN IOT -alustan

mahdollisuudet tulee ottaa huomioon startup-yrityksessä siinä vaiheessa, kun uutta SDx-alustaversiota ja tuotekehityspolkua suunnitellaan.

4.9 Euroopan yhtenäiset digitaaliset markkinat

Euroopan unionissa on 500 miljoonaa asukasta. Nyt vain 7 % pienistä ja keskisuurista yrityksistä tekee rajat ylittävää kaupankäyntiä. Kaupankäynnin esteiden poistamiseksi EU:n alueelle luodaan yhteiset digitaaliset markkinat (Digital Single Market, DSM) [148]. Tämän vuoksi on muun muassa muodostettu yhteenliittymä ”The Alliance for Internet of Things Innovation” (AIOTI) [149]. Standardoinnista huolehtii European Telecommunications Standards Institute (ETSI) [151, 152] yhdessä European Committee for Standardizationin (CEN) ja European Committee for Electrotechnical Standardizationin (CENELEC) kanssa [150]. Digitaaliset sovellukset voidaan toteuttaa FIWARE-alustalla [154; 155].

4.9.1 Euroopan IoT-standardointi

Euroopan alueen kaikkien tuotteiden ja palvelujen turvallisuuden ja laadun standardisoinnista vastaa virallisesti kolme eurooppalaista standardisointijärjestöä: European Committee for Standardization (CEN), European Committee for Electrotechnical Standardization (CENELEC) ja European Telecommunication Standards Institute (ETSI). [150.]

ETSI eurooppalaisena telealan standardisointijärjestönä on aloittanut IoT-standardoinnin [152]. Standardit ja niihin liitetyt API-rajapinnat ovat oleellisia, koska IoT-laitteiden tulee kyetä yhteistoimintaan ja kommunikoidaan keskenään. Monet IoT-liiketoimintamallit pohjautuvat tiedon välittämiseen useiden laitteiden ja organisaatioiden välillä. [151.]

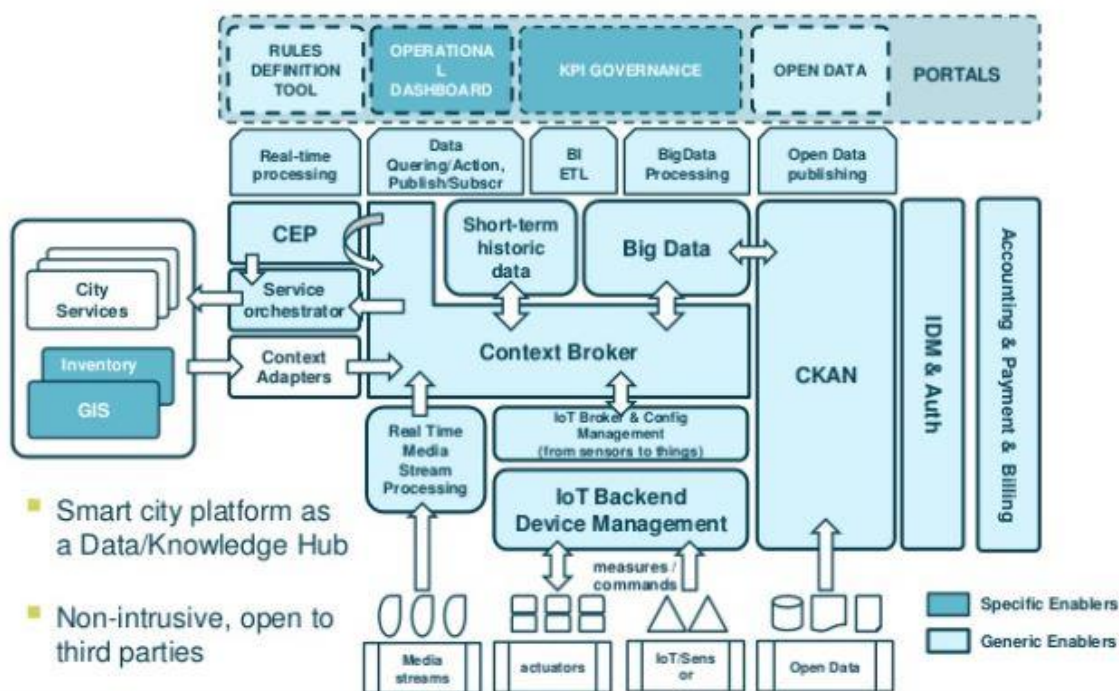
IETF globaalina internetprotokollien standardisointijärjestönä työskentelee yhdessä esimerkiksi Internet Protocol (IP) for Smart Objectsin (IPSO) ja ZigBee Alliance-, Open Mobile Alliance Lightweight Machine to Machine (OMA LWM2M)- ja Universal Plug and Play (UPnP) -standardisointijärjestöjen kanssa luodakseen IoT:lle yhteneväiset data- ja informaatiomallit [153].

4.9.2 FIWARE-alusta

Eurooppalaisista lähtökohdista kehitetty FIWARE-pilvipalvelinympäristöarkkitehtuuri noudattelee ETSlin standardointia, jolla on suuri liiketoiminnallinen merkitys FIWARE-ympäristöön tuotteita ja palveluja kehittäville yrityksille ja yhteisöille [154; 155].

4.9.3 FIWARE SmartCities

EU:n alueella käytetään FIWARE SmartCities -alustaa (kuva 46). Nimensä mukaisesti se on ensisijaisesti tarkoitettu kaupunkiympäristön palveluihin. EU:n ja Latinalaisen Amerikan alueen 31 kaupunkia seitsemästä maasta julkaisivat 15.3.2015 "Open & Agile Smart Cities" -aloitteen nopeuttaakseen yleisten standardien ja periaatteiden maailmanlaajuisia SmartCities-alustan käyttöönottoa. [154.]



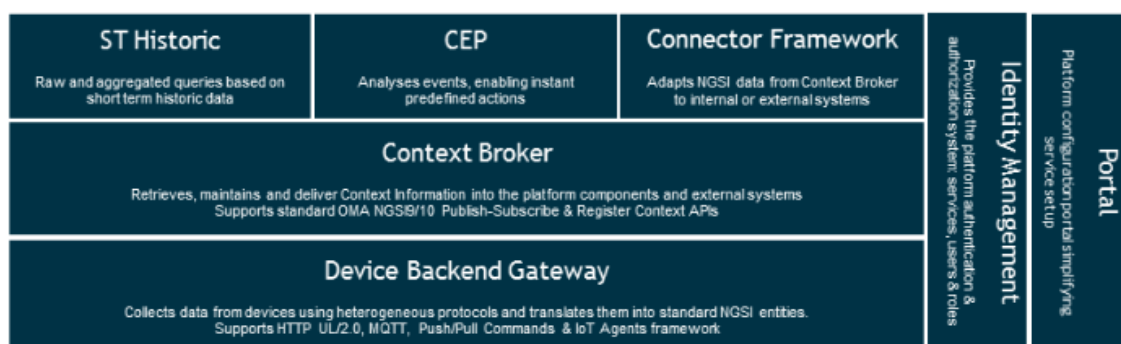
Kuva 46. FIWARE Smart City -alusta [157].

Suomesta SmartCities -kehitystyössä ja -käyttöön otossa ovat mukana Helsinki, Espoo, Vantaa, Tampere, Oulu ja Turku. Kaupungit voivat ottaa käyttöönsä avoimen Comprehensive Knowledge Archive Network (CKAN) -rajapinnan, jonka kautta sovelluskehittäjät pääsevät hyödyntämään kaupunkien tietokantoja ja rakentamaan IoT-sovelluksia. Sovelluskehittäjät voivat hyödyntää myös Next Generation Service Interfaces (NGSI) API -rajapintoja mahdollisten IoT-laitteistojen ohjauksessa. Kun sovellus on tehty yh-

teisten avoimien standardien ja periaatteiden mukaisesti yhdelle kaupungille, se on skaalattavissa kaikille SmartCities-alustaa käyttäville kaupungeille. Tämä mahdollistaa taloudellisesti kannattavan sovelluskehitystyön. [154.]

4.9.4 FIWAREn IoT-pino

FIWAREn IoT-pino on kuvassa 47. Samat komponentit ovat näkyvissä myös edellisessä SmartCities-alustan kuvassa 46.



Kuva 47. FIWAREn IoT-pino [158].

Kuvassa alimmalla tasolla eli yhteydestä fyysisiin sensoreihin huolehtii Device Backend Gateway, joka on toiselta nimeltään IDAS. Se tukee muun muassa standardeja Ultra-light 2.0 HTTP, Message Queuing Telemetry Transport (MQTT), Open Mobile Alliance Lightweight Machine to Machine (OMA LWM2M)/Constrained Application Protocol (CoAP) ja Push/Pull-komentoja sekä IoT Agents -kehystä, jolla kukin edellä mainittu standardirajapinta liitetään Context Brokeriin. FIWARE tarjoaa valmiiksi FIWARE-ympäristöön asennetun IDAS4:n, joka on FIWAREn projektien ja "testbed"-tarkoitukseen varattu ohjelma, johon sensorilaitteet voivat ottaa yhteyden. Jollei halua käyttää FIWAREn IDAS4:ää tai on tarve kehittää omaa IoTAgents-sensoriprotokollarajapintaa, IDAS-kehitysympäristön voi asentaa omaan CentOS VM -koneeseen ja käyttää sitä Fog-verkon laidalla vastaanottamaan esimerkiksi Raspberry Pi:ltä ja sen sensoreilta tulevaa tietoa. IDAS-kehitysympäristö sisältää muun muassa tietokantaohjelman MongoDBn sensoritietojen taltiointiin. On myös mahdollista asentaa FIGWAY-työkaluohjelma IDAS CentOS-VM -koneeseen, joka sisältää Context Brokerin ja IDAS:n ohjauskomentoja, joilla voi lähettää esimerkiksi simuloituja sensoriarvoja. Nämä ohjauskomennot on tehty Python-ohjelmointikielellä. FIWAREn kehittäjät suosittelevat käyttämään FIGWAY-työkaluohjelman komentoja FIWAREn IoT-ympäristöön.

tutustuessa. Toinen tapa Context Broker- ja IDAS-ohjauskomentojen lähettämiseen on käyttää niiden REST API -rajapintaa eli käyttää HTML:ään perustuvia lähetys- ja lukemiskomentoja. [158]. Tässä insinööriyössä käytettiin molempia edellä kuvattuja menetelmiä.

Seuraavalla tasolla on Context Broker eli sisällön välittäjätaso. Se tukee OpenMobileAlliancen standardia ”OMA NGSI9/10 Publish-Subscribe & Register Context API’s”, joka mahdollistaa sisällön välittämisen yleisen REST API -rajapinnan kautta. Jollei halua käyttää FIWAREn valmiiksi asennettua Context Brokeria, ohjelmiston voi asentaa VM-koneeseen tai PC:n VirtualBoxiin. [158].

Ylimmällä tasolla ovat ST Historic (STH) eli historiatiedon tallennus- ja hakutoiminnot ja Complex Event Processing (CEP), joka analysoi tapahtumia reaaliajassa. Se lähettää esimerkiksi sähköpostitse tiedon, jos lämpötila on noussut mitattavassa kohteessa yllä sallitun. Connector Framework on välikerros, joka perustuu Apachen Flumen tekniikkaan. Se välittää Context Brokerilta tai tulevan informaation FIWAREn ST Historic -tietokantaan ja myös muihin kolmannen osapuolen tuotteisiin, kuten esimerkiksi MySQL- ja MongoDB-tietokantoihin. Se tukee ”Hadoop Distributed File Systems” -tiedonvälitystekniikkaa. [158].

Kaikkia edellä mainittuja hallitaan web-pohjaisen portaalin kautta, ja Identify Managementillä huolehditaan todennuksista ja valtuuksista: palvelut, käyttäjät ja roolit. Nämä perustuvat OpenStack-pilvipalvelintekniikkaan. [158].

Loppukäyttäjälle tarkoitettu näkymä voidaan tehdä usealla tavalla. Voidaan käyttää FIWARE freeboard.io:n tarjoamaa ohjelmistoa [169], jolla voidaan helposti luoda esimerkiksi graafinen lämpötilamittari. Toinen tapa on käyttää FIWARE GE:n 2D- tai 3D-karttapohjalla näkyvä kokonaisuus, jossa voidaan näyttää esimerkiksi liikkuvat ajoneuvot ja IoT-sensoreiden välittämät tiedot. FIWARE 2D -karttapohjan toteutusesimerkki SmartSantander [159] on yksi Euroopan laajimmista SmartCity-toteutuksista. FIWARE 3D -karttapohjan yksi toteutusesimerkki on Fortum Espoon kaukolämpöverkon monitorointi, jossa voidaan näyttää rakennuksien sisällä olevia sensoritietoja. [160.]

5 SDx-tuotekehitysympäristön toteutus

Ohjelmoitavan maailman mahdollistava SDx-tuotekehitysympäristö on hyvin laaja ja monipuolinen kokonaisuus, joka sisältää kymmeniä ohjelmistoja. SDx käsittää silti vain yhden osan digitalisaation monista eri teknologioista (Gartnerin kuva 38). Tämän työn tutkimusosuus oli alustavaa tutkimustyötä SDx-tuotekehitysympäristön valinnalle.

Työssä tutkittiin kahta erilaista yhteisöä ja niiden tarjoamaa tuotekehitysalustaa.

SDN/NFV-tutkimusosuudessa tuli esiin avoin NFV-yhteisö **OPNFV**. IoT-tutkimusosuudessa tuli esiin FIWARE-alusta. Näistä kahdesta soveltuvampi tämän startup-projektin SDx-tuotekehitysalustaksi on **FIWARE**-alusta. Startup-projektin jälkeen perustettavaksi aiotun yrityksen tarpeiden näkökulmasta OPNFV-alusta voisi tuoda enemmän mahdollisuuksia. OPNFV on lähinnä suurten ”carrier-grade”-tason telecom-yritysten käyttämä ja mahdollisesti soveltuvampi alusta silloin, jos tulevan yrityksen tuotekehityksen painopiste suuntautuu enemmän SDN/NFV-teknologiaan. Molemmissa alustoissa, niin OPNFV:ssä kuin FIWAREssa, SDN/NFV tulee olemaan keskeinen tekijä.

5.1 SDx-ympäristön valinta

5.1.1 Pilvipalvelinympäristön valinta

Pilvipalvelinympäristöjen vertailuissa avoimuus ja hinta ratkaisivat valinnan OpenStackin hyväksi, joka tukee startup-yrityksen tulevaa ohjelmistokehitystyötä. Aiemmin vertailtiin OpenStack-teknologiaan pohjautuvia pilvipalvelun tarjoajia Platform9:ää, Nebulaa ja FIWAREa. Näistä kolmesta valittiin FIWARE ja Nebulan pilvipalvelinympäristöt.

OpenStack-teknologiaa käyttää myös avoin NFV-yhteisö OPNFV, ja se oli yksi vaihtoehtoista FIWAREn ohella, mutta ei tullut valituksi. OPNFV:n ja OpenStack-yhteisön kautta aloitteleva startup-yritys pääsee kehittämään täysin avoimen yhteisön kanssa telecom SDN/NFV-tuoteperhettään kaupalliseksi tuotteeksi. Monesti tätä kautta edenneet SDN/NFV-yritykset tulevat lopulta myydyksi isommalle telecom-yritykselle, joka ottaa uuden SDN/NFV-tuoteperheen osaksi omaa tuoteportfoliota ja säästyy itse riskialttiilta ja työläältä startup-vaiheelta. Tästä hyvänä esimerkkinä voi pitää Vyattaa, jon-

ka Brocade osti vuonna 2012. Vyattasta ja senjälkeisestä avoimesta VyOSista on kerrottu lisää luvussa 3.11.3 Quagga-kytkinohjelmisto.

FIWARE-yhteisön alustassa on laaS-pilvipalvelun lisäksi suurelta osin kaikki pilvipalvelin-, SDN/NFV- ja IoT-teknologioiden käytössä tarvittavat ratkaisut, joten se oli vertailussa olleista OpenStack-alustaa käyttävistä OPNFV- ja FIWARE-yhteisöistä soveltuvin startup-yrityksen SDx-kehitysympäristöksi. FIWAREn etuihin kuuluvat sen avoimuus, laajuus, tuki ja edullisuus. Euroopan unionin seitsemäs kehysohjelma osarahoitti FIWARE-yhteisöä kesäkuuhun 2016 asti. Siitä eteenpäin FIWAREa ylläpitää FIWARE-yhteisön jäsenet ja sen jäsenyritykset, joita on lukuisia. FIWARE on itsenäinen avoin yhteisö, jonka jäsenet ovat sitoutuneet rakentamaan julkisen ja maksuttoman sekä kestävän ekosysteemin. Tämä ekosysteemi tarjoaa muun muassa pienille ja keskisuurille yrityksille hyvän toimintaympäristön ja mahdollisuuden laajentua Euroopan ulkopuolelle FIWARE-yhteisön mukana. FIWARE-alusta sisältää muun muassa OpenStack-pilvipalvelinympäristön, QEMU/KVM-virtualisointiympäristön ja OFNIC-kontrollerin, joka on laajennus OpenDaylight-kontrolleriin. [154.]

FIWARE-alustan valintaan vaikutti lisäksi Euroopan laajuisen SmartCities-plattformin tuki, jota monien Euroopan kaupunkien, kuten Helsingin, Espoon ja Vantaan, avoimet rajapinnat tukevat. FIWAREn valintaprosessissa hyväksytyksi tulleet startup-projektit saavat enintään viisi OpenStack VM -konetta käyttöönsä ilmaiseksi yhdeksän kuukauden ajaksi. [154.]

Myös kaupallinen OpenStack-teknologiaan pohjautuva Nebula-hybridipilvipalvelu täytti vaatimukset ja tuli siten valituksi Fog-verkon rakentamisen ansiosta. Suomalaisena yksityisenä palveluntarjoajana se pystyy tukemaan startup-yrityksen toimintaa Suomessa ja pääkaupunkiseudulla. Jatkoa ajatellen Nebula Developer PaaS -palvelu tarjoaa ketterän ohjelmistokehityksen Micro-Services- ja Docker-teknologioin [163]. Nebulan palvelussa on mukana muun muassa RedHatin OPENSIFT [164], jonka voi integroida osaksi versionhallintaa. Sillä voi määritellä build-säännöt, Docker imaget ja Docker Build -automatisoinnin. Siinä on mukana myös Kubernetes, joka on tarkoitettu ohjelmistojulkaisujen optimointiin ja jolla voi joustavasti skaalata Docker-kontainerit. [165.]

Tulevaisuuden pilvialustalaajennuksina FIWAREn ja Nebulan OpenStackiin lisäksi voisi käyttää megapilviä, kuten Amazon Web Services, Microsoft Azure ja Google

Cloud Platform. Näiden alustojen yhteisen skaalauksen hallinnassa voisi käyttää kaupallista VMware Cross-Cloud –arkkitehtuuria. [110.]

Virtualisointiympäristö

Virtualisointiympäristön valintaan vaikutti paljon se, mikä on parhaiten tuettu valitussa FIWAREn OpenStack-pilvipalvelinympäristössä. OpenStack on parhaiten suunniteltu tukemaan QEMU/KVM-virtualisointiympäristöä, ja siten luonnolliseksi valinnaksi tuli QEMU/KVM. OpenStack mahdollistaa myös muiden virtualisointialustojen käytön.

Linux-jakelu

Ylläpidon kannalta perustelluin pitkän aikavälin valinta yrityksen Linux-jakeluksi on RedHat Enterprise Linux (RHEL). Startup-projektissa voidaan käyttää CentOS- ja Ubuntu-ilmaisjakeluita maksullisen RedHat-version sijaan. Raspberry Pi -laitteistoissa valittiin käytettäväksi Raspbian-, CentOS-, Ubuntu-, Ubuntu-Mate- ja Microsoft 10 IoT -käyttöjärjestelmiä. Näiden käyttöjärjestelmien valintaan vaikutti myös se, että niitä oli käytetty tässä insinööriyössä avoimien ohjelmistojen käyttöjärjestelmäalustoina.

DevOps-työkaluympäristöt

IoT-palveluohjelmistot tulee kyetä julkaisemaan maailmanlaajuisesti nopeassa tahdissa, luotettavasti ja pienellä henkilöstömäärällä. Tähän tarvitaan DevOps-työkaluohjelmistojä. Tulevaa startup-yrityksen jatkokehitystä ajatellen DevOps on yksi tärkeimmistä osa-alueista. Pilvipalvelinympäristöt ovat siirtymässä "host-centric"-infrastruktuurista "container-centric"-infrastruktuuriin. Tuotekehitysympäristön tuottamia telecom- ja IoT-palveluohjelmistojä on luontevaa suorittaa kontainereissa [162].

Kontaineriksi valittiin Docker container -ohjelmisto tukemaan Linux- ja myös Windows-pohjaista tuotekehitystyötä. Esimerkiksi Linux Container (LXC) -ohjelmisto tukee vain Linuxiin pohjautuvia ohjelmistojä. Kontainereiden etuihin kuuluu muun muassa se, että IoT-palveluohjelmistojä voidaan suorittaa eri ympäristöissä tarvitsematta muuttaa sitä ympäristöön sopivaksi. Se mahdollistaa IoT-palveluohjelmiston sujuvat siirrot tuotekehityks-, testaus- ja tuotantoympäristöjen välillä. Kontainer-infrastruktuurin orkesterointiin valittiin Kubernetes-ohjelmistoa [165]. Microservices PaaS -palvelun toteuttamiseksi valittiin tuotantokäyttöön soveltuva kaupallinen RedHatin OPENSIFT, joka

mahdollistaa koko ympäristön automaattisen luonnin pilvi- ja päätelaiteympäristöön [164]. Vaihtoehtoinen microservices PaaS -ympäristön toteuttaja voisi olla suomalainen avoimen lähdekoodin Kontena, joka lupaa ympäristön rakentamiseen käytetyn ajan vähenevän viikoista 15 minuuttiin [166; 167].

Muita DevOps-käyttöön mahdollisesti otettavia työkaluohjelmistoja ovat muun muassa Vagrant, Ansible, Chef, Puppet, Salt ja Jenkins. Versionhallintaohjelmaksi valittiin yleisyytensä vuoksi GitHub.

5.1.2 SDN/NFV-ympäristön valinta

SDN/NFV:n ohjausosaksi (Control Plane) valittiin avoimen ohjelmiston SDN-alusta OpenDaylight ja sen ONOS- ja OFNIC-kontrolleriympäristöt. Pilvipalvelinympäristöksi valitun FIWAREn VM-koneiden VLAN-liikennettä ohjaa FIWAREn OFNIC-kontrolleri. ONOS-verkonhallinta- ja kontrolleriohjelmistolla voidaan hallita OpenFlow-protokollayhteydellä vaihtoehtoisesti erilaisia kontrollereita, minkä mahdollistaa aiemmin tutkittu SDN OpenFlow -yhteyden virtualisointi. ONOS-verkonhallintaohjelmassa on vakiona ONOS:n oma Niciran-tekniikkaan pohjautuva kontrolleri. Jos myöhemmin on tarvetta, IoT-ympäristöä ohjaava ONOS:n Nicira-kontrolleri voidaan vaihtaa FIWAREn OFNIC-kontrolleriin. Kontrollerin vaihdokselle ei kuitenkaan ole näköpiirissä erityistä tarvetta.

Datan siirto-osaksi (Data plane) valittiin Open vSwitch (2.4.0) -ohjelmisto, joka tukee OpenFlow'n versioita v1.0, 1,2 ja 1.3. Vielä julkaisematomaan ja keskeneräiseen ONF:n määrittelemään OpenFlow 2.0 -versioon tutustuttiin syvällisesti teoriatasolla luvussa 3.7.2 OpenFlow 2.0 -protokolla - P4 ja POF Framework, mutta se ei tullut valituksi FIWARE-alustaan. P4 ja POF ovat lupaavassa kehitysvaiheessa, ja merkittäviä alan toimijoita on liittynyt niiden kehittämiseen. Esimerkiksi kuluttaja- tai yrityskäyttöön suunniteltujen tuotteiden suunnittelu tätä ympäristöä hyväksi käyttäen ei ole vielä kovin järkevää ennen kuin ONF-työryhmän OpenFlow 2.0 -standardointi on saatu päätökseen ja ne ovat yleisessä käytössä.

Aiemmin tutkittua Quagga-pohjaisia Brocade Vyatta- ja VyOS-käyttöjärjestelmiä ei valittu, mutta myöhemmässä startup-yrityksen SDN-reititintuotekehitystyössä niitä voidaan hyödyntää.

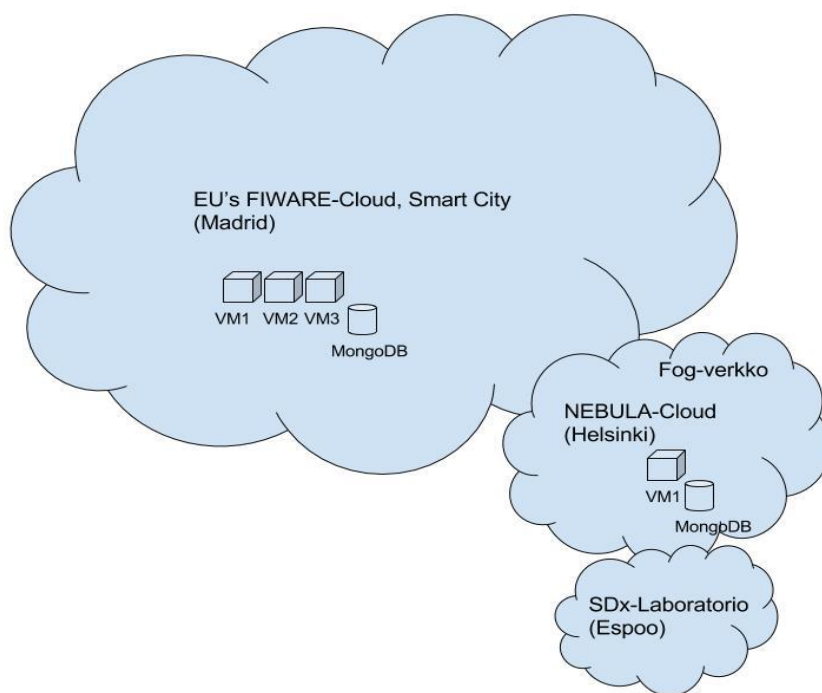
Kaupallisia SDN-ympäristöjen toimittajien tuotteita ei tullut valituksi, mutta tutkimus oli hyödyksi kaupallisten SDN-ympäristöjen markkinoiden ymmärtämisessä.

5.1.3 IoT-ympäristön valinta

IoT-ympäristöksi valittiin FIWARE-pilvipalvelinympäristön käyttämä ja ylläpitämä IoT-ympäristö. Valintaan vaikutti ympäristön täysi avoimuus ja esimerkiksi IoT SmartCities-palvelualusta, jolla tulevat startup-yrityksen IoT-palvelut voidaan parhaiten toteuttaa. Vaihtoehtoisia IoT-ympäristöjä tutkittiin ja otettiin käyttöön muun muassa Microsoft 10 IoT-ympäristö ja ThingSpeak IoT -ympäristö MATLAB-tuen vuoksi. Muita tunnettuja IoT-ympäristöjä tarjoavat myös Google, Amazon ja IBM. Nämä kaikki ovat kuitenkin enemmän tai vähemmän suljettuja ympäristöjä ja yhden yrityksen hallinnoimia, ja niiden kehitykseen eivät ohjelmistoyritykset pääse kovin syvällisesti vaikuttamaan.

5.1.4 SDx-ympäristön topologia kuvaus

Tehtyjen valintojen perusteella muodostettiin SDx-ympäristön ylätasen topologia kuvaus (kuva 48).



Kuva 48. SDx-ympäristön topologia.

SDx-ympäristön topologiakuva kuvaa toteutettavaa ympäristöä, jonka tarkempi suunnittelu tehtiin seuraavaksi. Suunnittelu pohjautui valittuihin tekniikoihin.

5.2 SDx-ympäristön suunnittelu

Startup-projektin SDx-tuotekehitysalustan ensimmäiseen versioon valitut pilvipalvelin-, SDN/NFV- ja IoT-teknologiat otetaan mahdollisimman laajasti käyttöön.

Lisäksi tutustumistarkoituksessa tämän insinööriyön ulkopuolella voidaan rakentaa Microsoft Azure IoT -ympäristökokonaisuus ja ThingSpeak IoT -ympäristö graafisella käyttöliittymällä sensoritietojen näyttämiseksi loppukäyttäjälle. DevOps-työkaluja otetaan laajemmin käyttöön seuraavassa SDx-tuotekehitysalustaversiossa.

Seuraavaksi kuvataan suunnitelman pääkohdat.

5.2.1 Graafisen käyttöliittymän suunnittelu

Sensoritiedon näyttämiseksi voidaan käyttää FIWARE freeboardin **graafista käyttöliittymää** [169] (kuva 49, lohko 1). Vielä laajemassa tarkoituksessa voidaan käyttää FIWAREn 2D- ja 3D-karttaohjelmistoja sekä laajennetun todellisuuden Augmented Reality -ohjelmistoa.

5.2.2 Pilvipalvelinympäristön suunnittelu

Suunniteltiin **Fog-pilvipalvelinympäristö**, joka muodostuu FIWARE- ja Nebula-pilvipalvelimista.

FIWARE-pilvipalvelinympäristön suunnittelu (kuva 49, lohko 2)

Ensin osallistuttiin FIWAREn valintaprosessiin ja muodostettiin tili palvelujen rakentamiseksi. Tämän jälkeen FIWARE-Cloud OpenStack IaaS -alustalle rakennettiin PaaS-tuotekehitysympäristö, joka voi sisältää seuraavat VM-koneet: kaksi Ubuntu VM -koneita ja yksi CentOS VM -kone kytkettynä sisäiseen VLAN-verkkoon, joka on toteutettu FIWAREn omalla SDN OFNIC -kontrollerilla. VM-kone 1 on tuotekehityskäytössä, VM-kone 2 on testaus ja prototyyppikäytössä ja VM-kone 3 on tuotantoympäristön käy-

tössä. Tarpeen vaatiessa tätä ympäristöä on mahdollisuus laajentaa. VM-kone 1 sisältää ONOS-verkonhallinta- ja kontrolleriohjelmiston, joka ohjaa kentällä olevaa R-Pi-laitetta, joka sisältää SDN Open vSwitch -kytkimen. Kytkimen neljään Ethernet-porttiin on kytketty kiinteistön IoT-laitteistoja soveltuva määrä. Ne tulevat jatkossa välittämään FIWAREn IoT-alustan kautta sensoridataa FIWARE OpenStack VM -kone 2:lla toimivalle FIWAREn IoT-palveluohjelmistolle Orion ContextBrokerille ja sen MongoDB:lle. VM-kone 3:lla sijaiseva OFNIC-kontrolleri on varattu mahdollista myöhempää käyttöä varten.

Nebula-Cloud-pilvipalvelinympäristön suunnittelu (kuva 49, lohko 3)

Nebula-Cloudiin perustetaan tili ja asennetaan CentOS VM -kone ja sinne FIWARE IDAS -ohjelmisto sekä MongoDB tiedon väliaikaista taltiointia ja analysointia varten. Sieltä välitetään vain tarpeellinen sensoritieto FIWARE-Cloudissa sijaitsevaan ContextBrokeriin. Tällä tavoin saadaan luotua Fog-network eli ”sumuverkko”, joka on levinnyt lähelle sensorilaitteita. Näin saadaan tiedonsiirron latenssi sensorilaitteilta IDAS-ohjelmistolle ja takaisin pysymään kohtullisena eli noin 10 ms:ssa. Latenssia saadaan pienennettyä aina noin 1 ms:iin saakka levittämällä ”sumuverkkoa” vielä lähemmäksi sensorilaitteita esimerkiksi samoihin kiinteistötiloihin saakka. Pientä latenssia tarvitaan muun muassa älyliikenteen Mobility As A Service (MaaS) -sovelluksissa. Autoliikenteen ohjauksessa pieni latenssi ja luotettavuus ovat erityisen tärkeitä tekijöitä. Tulevaisuuden autoliikenteen sensoriverkoissa voidaan käyttää Wirepas-sensoriverkkotekniikkaa, jolla sensoritieto voidaan siirtää suoraan autosta autoon ja näin latenssi saadaan hyvin pieneksi. [142.]

5.2.3 SDN/NFV-ympäristön suunnittelu

SDx-laboratorioympäristössä sijaitseva **SDN/NFV Coursera -kurssiympäristö** voidaan asentaa PC:lle. Se mahdollistaa SDN/NFV-ohjelmistotuotteiden suunnittelun virtuaalisessa ympäristössä. Erilaiset SDN/NFV-verkkotopologiat ja NFV ”middlebox” -ohjelmistot on helppo toteuttaa ja testata, minkä jälkeen ohjelmistot voidaan siirtää todelliseen SDx-ympäristöön. Suoritetaan Princetonin yliopiston SDN/NFV-kurssi ja asennetaan sen ohjelmistoympäristö omalle PC:lle, kurssin ohjeiden mukaisesti. Kurssiympäristökokonaisuus käynnistetään yhdellä Vagrant-komennolla, joka on ”vagrant up”. Kurssilla voidaan suorittaa muun muassa NFV ”middlebox” -ohjelmistoharjoituksia ja P4-ohjelmointikieliharjoituksia.

SDN/NFV-kytkinympäristön suunnittelu Raspberry Pi -laitteistolle (kuva 49, lohkot 2, 4 ja 5)

Pilvestä käsin ohjattava SDN/NFV-kytkin ja IoT-yhdyskäytävä (gateway, GW) asennetaan Raspberry Pi -laitteistoille lähelle sensorilaitteistoja. Käyttöjärjestelmä ja ohjelmit tulee valita sen mukaisesti, mitä valittu FIWARE IoT -alusta vaatii toimiakseen.

5.2.4 IoT-ympäristön suunnittelu

IoT-ympäristön suunnittelu (kuva 49, lohkot 1, 2, 3, (4), 5, 6, 7 ja 8)

Monissa IoT-ympäristöissä kaikki sensoridata viedään pilvipalvelimeen, missä sitä analysoidaan ja tehdään sen pohjalta tarvittavat toimenpiteet. Tämä kuormittaa tarpeettomasti tietoliikenneverkkoa ja pilvipalvelinta ja aiheuttaa ylimääräisiä ylläpitokustannuksia [149]. Esimerkiksi jos pilvipalvelimien IoT-ympäristön laskutus perustuu ”Pay-As-You-Go” -laskutukseen, kuten Microsoft Azure IoT -ympäristössä, tällöin pilvipalvelimelle tulevien sensoriviestien suuri lukumäärä vaikuttaa laskun suuruuteen [172].

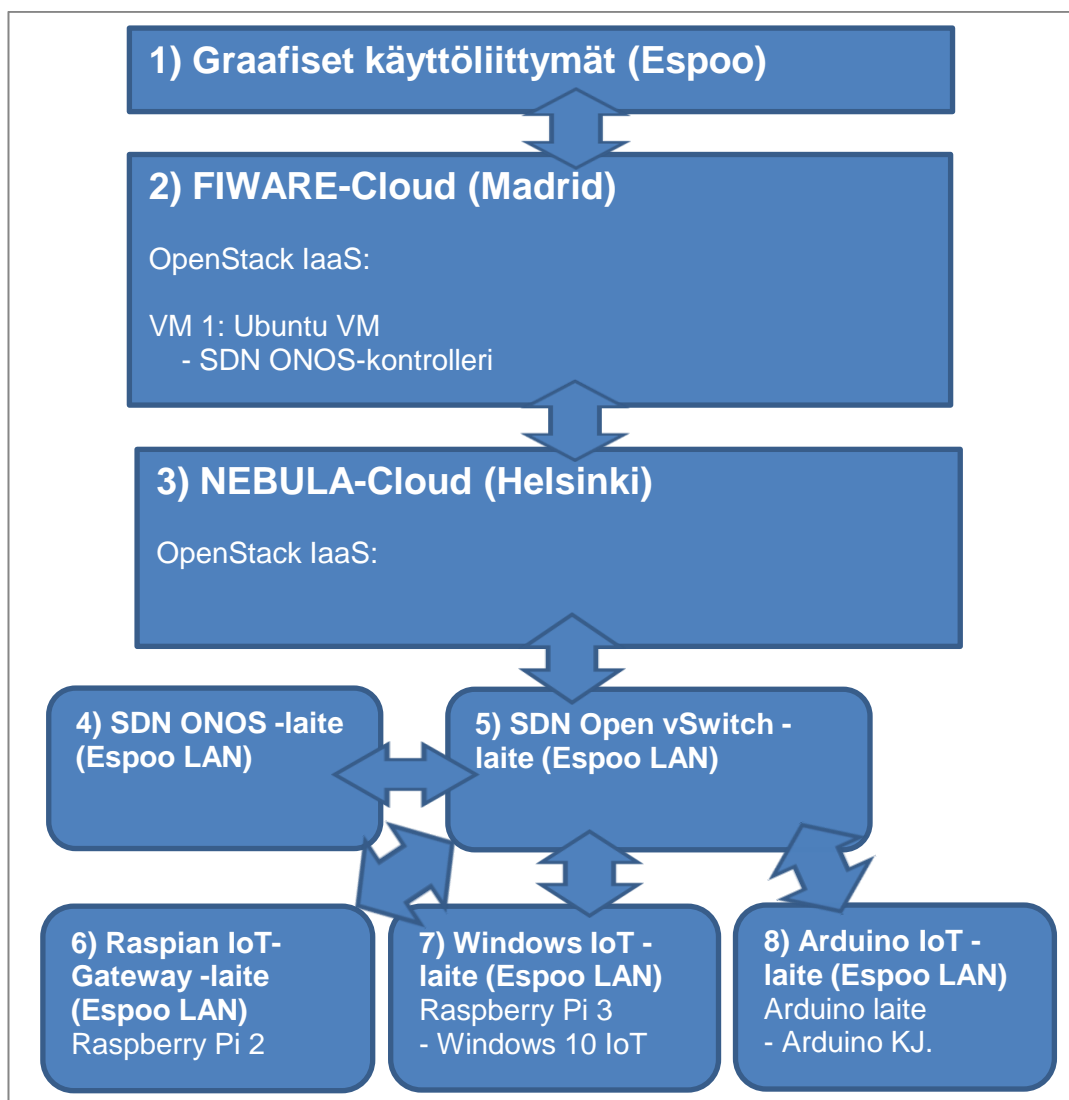
FIWAREn IoT-ympäristön tulisi käyttää niin sanottua Fog-networking -menetelmää, jossa sumupilvi ulottuu lähelle loppukäyttäjää. FIWAREn sumupilven sisällä käytetään FIWAREn NGSI9- ja 10-kommunikointiprotokollaa. Pilven reunalla lähellä IoT-sensoreita suoritetaan sensoritiedon esikäsittelyä, ja FIWAREn pilvipalvelimessa sijaitsevalle ContextBrokerille lähetetään vain tarpeellisin tieto. Esimerkiksi pilven reunalla tutkitaan, onko lämpötila laskenut kiinteistössä ali tai yli tietyn raja-arvon. Jos on, vain tämä ja lämpötilamuutostieto välitetään pilvipalvelimelle, joka antaa tarvittaessa hälytyksen loppukäyttäjälle. Pilvipalvelimessa sijaitsevassa IoT-palvelun käyttöliittymästä loppukäyttäjä pääsee ohjaamaan laitteistojaan ja seuraamaan esimerkiksi pitkän aikavälin lämpötilan historiatietoja ja järjestelmän antamia hälytyksiä.

Startup-projektin SDx-tuotekehitysympäristön kehittämisen kannalta itse sensoritietolähteellä (lämpötila, kosteus jne.) ei ole merkitystä. Sensoritiedon välittämisessä antureilta langattomasti tai langallisesti sumupilven reunalle voidaan käyttää useammanlaista tekniikkaa, kuten esimerkiksi Ultralight 2.0 Hypertext Trasfer Protocol (HTTP)-, Message Queuing Telemetry Transport (MQTT)- ja Open Mobile Alliance (OMA) Lightweight Machine to Machine (LWM2M) / Constrained Application Protocol (CoAP) -protokollia. Langattoman sensoriverkon valinnassa tulee ottaa huomioon muun muassa

virrankulutus eli se, onko kyseessä verkkovirta- vai paristokäyttöinen laite ja mikä on tiedonsiirron määrän tarve. Valinnassa tulee ottaa huomioon myös kuinka suuri IoT-liikenteen latenssi voi olla ja kuinka laaja alue kyseisen sensoriverkon tulee kattaa. Hyvin tärkeitä ovat tietoturvakysymykset. Sensoriverkkotekniikka tulee valita tarjottavaksi aiotun palvelun perusteella. Tässä insinöörityössä ensimmäiseksi sensoriverkkotekniikaksi valittiin tavallinen HTTP- ja Ultralight 2.0 http -protokolla. Niiden etuihin kuuluu helppo käyttöönotto ja soveltuvuus tämän insinöörityön IoT-osuuden toteuttamiseksi, koska siinä sensoritieto välittyy tavallista LAN-verkkoa ja yleistä IP-verkkoa hyödyntäen aina Nebula-Cloudissa sijaitsevalle FIWARE IDAS Ultralight 2.0 http -agentille asti. Sensoritieto välittyy siitä eteenpäin FIWAREn sumupilvessä NGSI 9/10 -protokollilla FIWARE-Cloudissa olevalle ContextBrokerille asti ja eteenpäin freeboardin graafiselle käyttöliittymälle.

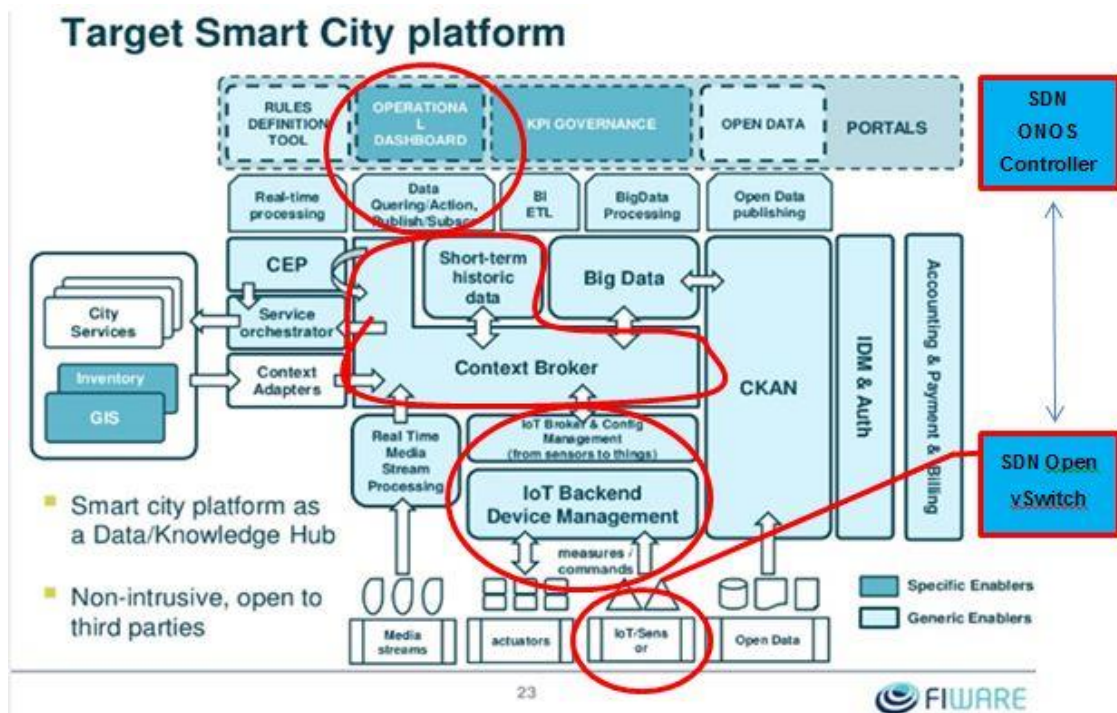
5.2.5 SDx-ympäristön arkkitehtuuri

Toteutettava SDx-ympäristö (kuva 48, 49) perustuu FIWARE Smart City -alustaan (kuva 50). Etelä- ja Keski-Euroopan alueella sijaitsevaa FIWARE-pilveä on tässä työssä laajennettu Suomen pääkaupunkiseudulle asti. Tämä laajennus on nimeltään Fog-verkko (Nebula-Cloud, kuva 49, lohko 3). Toinen laajennus on start-up-projektin laboratoriotiloissa Espoossa, joka on ADSL-modeemiyhteyden päässä ja LAN-verkkoon liitetty SDN/NFV-kytkin (kuva 49, lohkot (2), 4 ja 5) sekä IoT-laitteistot ja sensorit (kuva 49, lohkot 6, 7 ja 8).



Kuva 49. SDx-ympäristön lohkokuva.

Kuvassa 50 näkyvään FIWARE Smart City -alustan arkkitehtuurikuvaan on rajattu punaisella tässä työssä toteutetut lohkot. Toteutukseen on myös lisätty SDN Open vSwitch (kuva 49 lohkot 2 ja 5). Se sijaitsee alimmaisten IoT-laitteistojen yläpuolella ohjaten sensoreilta tulevaa informaatiiovirtaa. Jatkokehitystä ajatellen tämä mahdollistaa IoT-liikenteelle aiemmin luvussa 3.1.5 NFV - verkkotoimintojen virtualisointi esitetyt "middlebox"-in-toiminnot. Ne voivat sijaita lohkossa 3 Nebula-Cloudin CentOS VM - koneessa tai vielä lähempänä IoT-sensorilaitteita lohkossa 4 Raspberry Pi -laitteessa, jos on tarve pienelle latenssille. Näitä toimintoja voi hyödyntää monissa digitalisaation palveluissa, kuten älyliikenteen MaaS-palveluissa.



Kuva 50. FIWARE Smart City -alusta. Työssä toteutetut SDx-ympäristön lohkot on merkitty punaisella (vrt. kuva 49).

5.3 SDx-ympäristön toteutus

5.3.1 Graafisen käyttöliittymän toteutus

Graafinen käyttöliittymä on kuvassa 49 lohko 1. FIWARE freeboardin asennusvaihtoehtoja on kolme: 1) asennus paikallisesti omalle esimerkiksi Raspberry Pi -laitteelle, 2) käyttö freeboard.io:n verkkosivujen kautta luomalla tili omilla tunnuksilla, jolloin asennusta ei tarvitse tehdä ja 3) asennus omalle web-palvelimelle, esimerkiksi nodejs:lle. Näistä kolmesta toteutettiin vaihtoehdot 1 ja 2 ja konfiguroitiin ympäristö siten, että freeboardin graafisella näytöllä näkyy sensoritietoa näyttävät mittarit. Asennusohjeet löytyvät freeboard.io:n verkkosivuilta [169].

2D- ja 3D-karttapohjia tai laajennettua todellisuutta ei pidetty tässä vaiheessa tarpeellisenä toteuttaa. Smartsantanderin esimerkkitoteutuksesta [ks. lähde 159] saa hyvän käsityksen laajasta IoT-karttapohjatoteutuksesta ja sen mahdollisuuksista.

5.3.2 Pilvipalvelinympäristön toteutus

FIWARE-pilvipalvelinympäristön toteutus (kuva 49, lohko 2)

Pilvipalvelinympäristö FIWARE-Cloud asennettiin FIWAREn ohjeistuksien mukaisesti [154]. Seuraavaksi asennettiin kaksi Ubuntu-koneella ja yksi CentOS VM -kone FIWARE-Cloudin omista VM-konevaihtoehdoista eli kuvakkeista (image). ContextBroker-ohjelmisto oli integroitu valmiiksi CentOS-kuvakkeeseen, joten sitä ei tarvitse myöhemmässä IoT-toteutuksessa asentaa. FIWARE-Cloudissa OFNIC-kontrolleri tuli automaattisesti käyttöön muodostettaessa OpenStackiin VM-koneet. Se muodosti myös VM-koneet yhdistävän VLAN-verkon. Yksi VM-kone kytkettiin julkiseen IP-osoitteeseen. OpenDaylight sisältää myös ONOS-verkonhallintajärjestelmän [71], jota käytettiin tämän insinööriyön käytännön SDN/NFV-kytkinösuudessa sen skaalautuvuuden ja edistyksellisyysden vuoksi. Luotiin palomuuriryhmä (Security Groups) ja avattiin ympäristön käyttämät portit. VM-koneiden etäyhteyksiä varten luotiin avainpareja (Keypairs) ja otettiin etäyhteydet käyttöön.

Nebula-Cloud pilvipalvelinympäristön toteutus (kuva 49, lohko 3)

NEBULA-Cloud asennettiin Nebulan ohjeistuksen mukaisesti [23]. Asennus sujui samalla tavoin kuin FIWARE-Cloud, koska molemmat ovat OpenStack-pohjaisia IaaS-pilvipalvelinympäristöjä. Seuraavaksi asennettiin yksi CentOS VM -kone Nebula-Cloudin omista VM-konevaihtoehdoista eli kuvakkeista (image). Luotiin palomuuriryhmä (Security Groups) ja avattiin ympäristön käyttämät portit. CentOS VM -koneen etäyhteyttä varten luotiin avainpari (Keypairs) ja otettiin etäyhteys käyttöön.

5.3.3 SDN/NFV-ympäristön toteutus

SDN/NFV Coursera-kurssiympäristön toteutus PC:lle (kuva 51)

PC-ympäristöön asennettiin Princetonin yliopiston Coursera-kurssilla käytetyt ohjelmistot. Ennen Vagrant-ohjelman asennusta Mininet-verkkoemulaattoria varten tulee asentaa PC:lle Oracle VM VirtualBox [98]. Tämä tehdään Mininet-ohjeistuksen mukaisesti [86]. Vagrant-ohjelmiston avulla käynnistetään yhdellä "vagrant up" -komennolla kurssiympäristöohjelmistot, kuten esimerkiksi Mininet-ympäristö. Tarkat yksityiskohtaiset

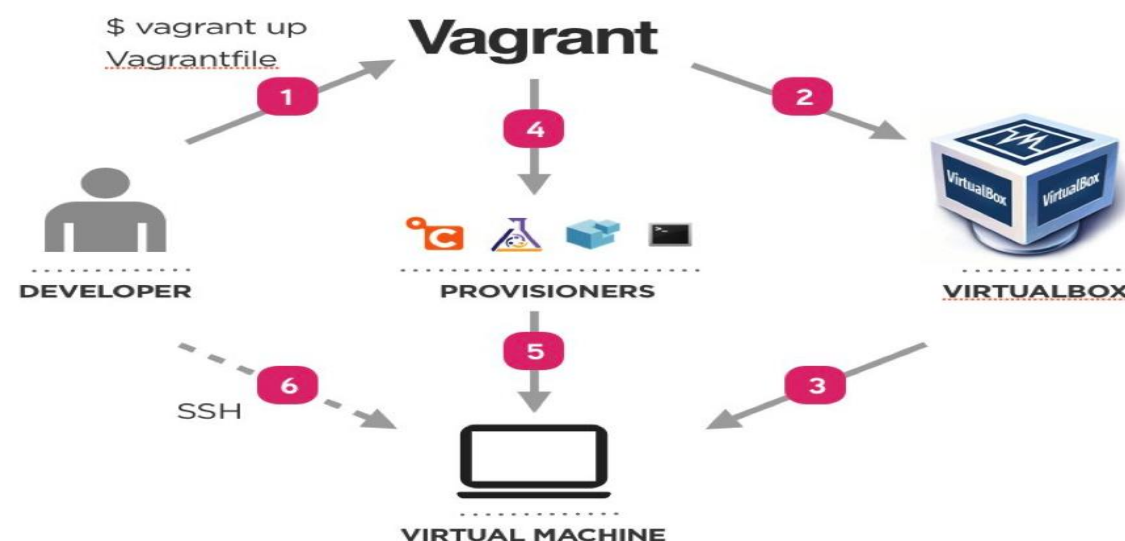
Vagrant-asennusohjeet löytyvät Princeton yliopiston Coursera-kurssimateriaalista [4]. Seuraavassa esitetään asennuksen pääkohdat ja Vagrant-ympäristön toimintaperiaate.

Tuotekehitysympäristö on helppo perustaa mihin tahansa tietokoneeseen (Windows, Macintosh, Linux) siten, että asennetaan ensin kurssilla käytettävät ohjelmistot ja käynnistetään Git Bash -terminaali-ohjelma [4]. Kaikki kurssin sisältämät ohjelmistot, kuten Mininet, on tallennettu GitHubiin [5]. PC:ssä annetaan komento:

```
git clone https://github.com/PrincetonUniversity/Coursera-SDN
```

Komennolla ohjelmistot saadaan kopioitua omalle pöytäkoneelle. Vagrantfile-tiedostossa määritellään kaikki VM-koneessa käynnistettävät ohjelmat, kuten esimerkiksi Mininet-ohjelma. Käynnistetään Vagrant-ohjelma `vagrant up` -komennolla. Käynnistyksen jälkeen otetaan Putty-ohjelmalla Secure Shell (SSH)-yhteys X11-ominaisuus valittuna oman VM-koneen IP-osoitteeseen (esimerkiksi 192.168.0.100). Yhteyden muodostuksen jälkeen avautuu näkymä SDN-kurssiympäristön hakemistoihin.

Kuvassa 51 on Vagrant-ympäristön käynnistysprosessi.



Kuva 51. Vagrant-ympäristön käynnistys [4].

Yleisesti ottaen Vagrant on asennettavissa erilaisiin virtuaaliympäristöihin esimerkiksi paravirtualisointi Oracle VirtualBox ja VMware Playeriin ja täysvirtualisointi VMware ESXi ja Hyper-Vn. Vagrantfile-tiedostossa voi määrittää myös muun muassa shell

skriptejä sekä Chef- ja Puppet -ohjelmistoja, joiden avulla voi automaattisesti asentaa ja konfiguroida käynnistettäviä ohjelmistoja. [4.]

Courseran kurssiympäristössä suoritettiin muun muassa Mininet-topologia-harjoituksia ja "middlebox"-hajoituksia, kuten palomuuuri ja kuormanjako, sekä ohjelmointiharjoituksia Python-, Frenetic-, Pyretic-, Kinetic-, NetAsm- ja P4-ohjelmointikielillä POX- ja Ryu-kontrollereihin. Myöhemmässä startup-yrityksen tuotekehitysvaiheessa voidaan ottaa käyttöön Coursera-ympäristössä suunniteltuja NFV "middlebox" -ohjelmistoja ja siirtää niitä soveltuvin osin nyt rakennettuun SDx-ympäristöön.

SDN/NFV-kytkinympäristön toteutus Raspberry Pi -laitteistolle (kuva 49, lohkot 2, 4 ja 5)

Kytkenän toteutus tehtiin FIWARE-pilvessä sijaitsevalla ONOS-kontrollerilla (lohko 2) ja kahdella R-Pi-laitteistolla:

- VM-kone 1: SDN ONOS -kontrolleri [175], jossa on Ubuntu-MATE-käyttöjärjestelmä [170] (kuva 49, lohko 2)

tai

- SDN ONOS -kontrollerilaitteena [175] Raspberry Pi 3, jossa Ubuntu-MATE-käyttöjärjestelmä [170] (kuva 49, lohko 4)
- SDN Open vSwitch -laitteena [176] Raspberry Pi 3, jossa Ubuntu-MATE-käyttöjärjestelmä [170] (kuva 49, lohko 5). Raspberryn kolme sisäistä Ethernet-porttia kytkettynä ulkoisiin USB-liityntöihin, joissa USB/Ethernet-adapterit.

VM-kone 1:n SDN ONOS -kontrolleri ja R-Pi SDN ONOS -kontrollerilaitte toimivat vaihtoehtoisina SDN-kontrollereina R-Pi 2 -laitteen SDN Open vSwitch -kytkimelle. Kontrolleri voidaan vaihtaa antamalla R-Pi 2 -laitteessa komento `sudo ovs-vsctl set-controller mybridge tcp:<IP-address>:6633`. IP-osoiteeksi tulee valitun kontrollerin osoite.

5.3.4 IoT-ympäristön toteutus

IoT-osuuden toteuttamiseksi otettiin käyttöön FIWARE freeboardin graafinen käyttöliittymä (kuva 49, lohko 1) ja ContextBroker FIWARE-Cloudin CentOS VM -koneessa (kuva 49, lohko 2). ContextBroker oli jo ennalta asennettu CentOS VM -koneen asennuksen yhteydessä.

FIWAREn IoT-ympäristön laajentaminen Suomeen ja pääkaupunkiseudulle eli niin sanotun Fog-verkon rakentaminen edellytti FIWARE Orion IDAS -ohjelmistoympäristön rakentamista (kuva 49, lohko 3). Tähän tarkoitukseen valittiin pääkaupunkiseudulla toimivan Nebula-Cloudin OpenStack IaaS -pilvipalvelu. Sen päälle asennettiin CentOS 7 VM -kone [174] ja IDAS-ohjelmisto ohjeiden mukaisesti [154].

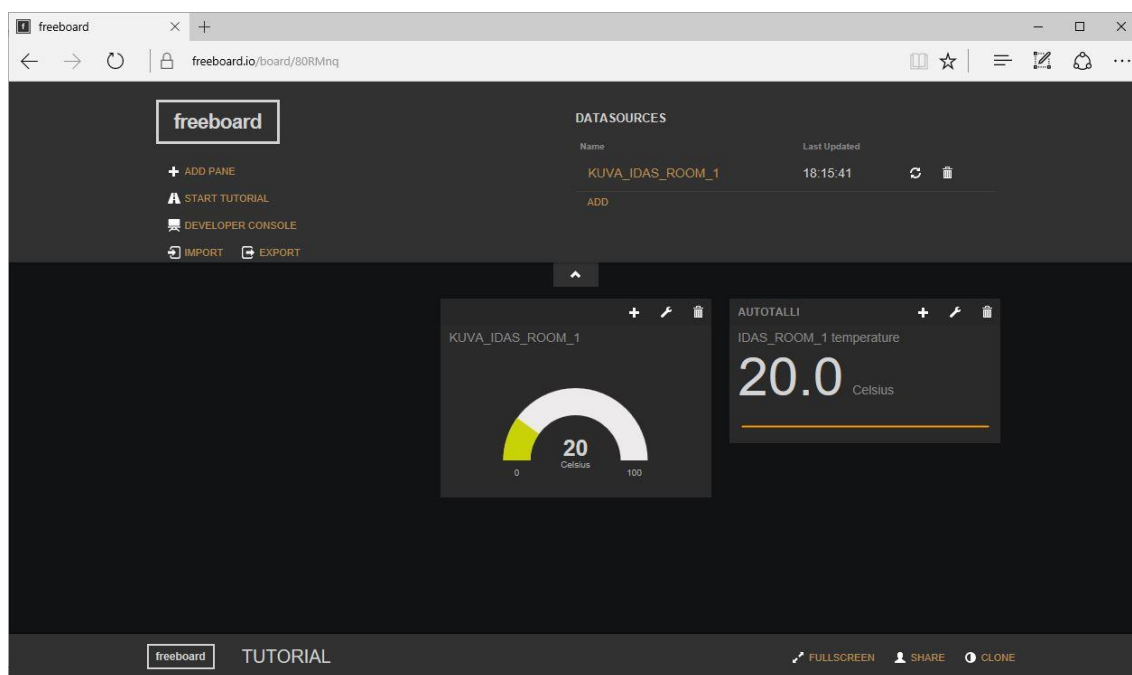
Startup-projektin SDx-laboratoriotiloihin asennettiin seuraavat Raspberry Pi -laitteistot:

- Raspbian IoT -yhdyskäytävä (gateway, GW) -laite [171; 177], joka toimii LAN-palvelimena (kuva 49, lohko 6). Sen kautta lähetetään keskitetysti kaikki IoT-sensoritieto pilvipalvelimille. Se lähettää IoT-sensoridataa Nebula-Cloudissa sijaisevalle FIWARE-IDAS-ohjelmistolle ja se edelleen FIWARE-Cloudissa sijaitsevalle FIWARE ContextBrokerille. Sensoritieto on sieltä luettavissa FIWARE freeboardin graafisella käyttöliittymäohjelmistolla. Sensoritietoa lähetetään myös Microsoft Azure IoT -pilvipalvelinympäristölle ja ThingSpeak.com:n web-pohjaiselle IoT-alustalle.
- Windows IoT -laite [172] (kuva 49, lohko 7), joka toimii LAN-asiakasohjelmana ja lähettää IoT-sensoridataa Raspbian IoT -yhdyskäytävälle, joka toimii palvelinkoneena.
- Arduino IoT -laite [173] (kuva 49, lohko 8), joka toimii LAN-asiakasohjelmana ja lähettää IoT-sensoridataa Raspbian IoT -yhdyskäytävälle, joka toimii LAN-palvelinkoneena.

5.4 SDx-ympäristön testaus

5.4.1 Graafisen käyttöliittymän testaus

Graafinen käyttöliittymä on kuvassa 49 lohko 1 ja kuvassa 48. Kytkenässä IoT-laitteisto lähettää sensoridataa FIWARE freeboardin graafiselle käyttöliittymälle (kuva 52) sekä Microsoft Azure IoT -ympäristölle ja web-pohjaiselle graafiselle ThingSpeak.com käyttöliittymälle. Kaikilla näillä alustoilla on havaittavissa IoT-sensoridatan saapuminen.



Kuva 52. FIWARE freeboard graafinen käyttöliittymä.

FIWARE freeboardin graafisen käyttöliittymän näkymä luvun 5.3.1 Graafisen käyttöliittymän toteutus, asennusvaihtoehto 2:n mukaisesti eli käyttö freeboard.io:n web-sivujen kautta. Freeboardin graafisen käyttöliittymän näkymässä on mittari ja tekstipohjainen lämpötilan näyttö.

5.4.2 Pilvipalvelinympäristön testaus

FIWARE-Cloud-pilvipalvelinympäristön testaus (kuva 49, lohko 2 ja 3)

Todettiin FIWARE-Cloud ja Nebula-Cloud -pilvipalvelinympäristöjen toiminta siten, että kyetään ottamaan Linux shell -terminaalista etäyhteys CentOS VM -koneeseen luotujen avainparien avulla.

```
pi@raspberrypi:~/FIWARE-Cloud/orion_Keypair_1_chmod $
ssh -i orion_Keypair_1.pem centos@<IP-address>
```

Nebula-Cloud-pilvipalvelinympäristön testaus (kuva 49, lohko 3)

Todetaan Nebula-Cloud-pilvipalvelinympäristön toiminta siten että kyetään ottamaan Linux shell -terminaalista etäyhteys CentOS VM -koneeseen luotujen avainparien avulla.

```
pi@raspberrypi:~/NEBULA-Cloud/IDAS_Keypair_1_chmod $
ssh -i IDAS_Keypair_1.pem centos@<IP-address>
```

5.4.3 SDN/NFV-ympäristön testaus

SDN/NFV Coursera-kurssiympäristön testaus PC:ssä

Tehtyjen harjoitustöiden testattiin Coursera-kurssiympäristön omilla vastauksenlähetyskripteillä kurssin tarjoajan palveluun.

SDN/NFV-kytkinympäristön testaus Raspberry Pi -laitteistolla (kuva 49, lohkot 2, 4 ja 5)

SDN ONOS -kontrollerilaitteella olevasta ONOS-kontrollerista käsin ”stop onos-app-fwd” -komennolla poistetaan SDN Open vSwitch-laitteistolla oleva L2-tason kytkentä. Tällöin havaitaan IoT-grafiikasta, että IoT-dataa ei enää välity FIWAREn freeboard-, Microsoft IoT -ympäristön ja ThingSpeak.com:n pilvipalveluihin. IoT-sensoritietona on FIWAREn freeboardissa käytetty lämpötilasensorin tuottamaa tietoa. Microsoft IoT-ympäristön ja ThingSpeak.com:n pilvipalveluihin on välitetty sensoritietoja kattavamasti muun muassa lämpötila-, kosteus-, liike- ja ovikytkin-sensoritiedot.

5.4.4 IoT-ympäristön testaus

Testattiin IoT-ympäristön sensoritietojen päivittymistä MongoDB:n tietokantoihin (kuva 49, lohkot 2, 3, 5 ja 6).

FIWARE ContextBrokerin MongoDB:n tietokannan testaus

Käynnistetään FIWARE ContextBrokerin MongoDB-ohjelma Linux shell -terminaalissa komennolla `mongo`

MongoDB:n komentotulkin komennolla otetaan tietokanta käyttöön: `use orion-howtoservice`

ja luetaan tietokannasta sensoreiden välittämää tietoa ja todetaan niiden päivittyminen:

```
> db.entities.find()
```

Tässä esimerkissä lämpötilatiedon arvo on 20.0 astetta.

FIWARE IDAS MongoDB:n tietokannan testaus

Käynnistetään FIWARE IDAS MongoDB -ohjelma Linux shell -terminaalissa komennolla `mongo`

MongoDB:n komentotulkin komennolla otetaan tietokanta käyttöön: `> use iotagentul`

ja luetaan tietokannasta sensoreiden välittämää tietoa ja todetaan niiden päivittyminen.

Tässä esimerkissä kaikki sensoritieto välitetään ilman välitallennusta IDAS MongoDB:n kautta eteenpäin ContextBrokerille: `> db.devices.find()`

Tässä esimerkissä lämpötilatiedon arvo on 0 astetta.

Jos pilvipalvelimen IoT-ympäristö olisi ollut maksullinen, kuten Microsoft Azuren ”Pay-As-You-Go” -palvelu, ylimääraistä IoT-liikennettä olisi ehkä syytä rajoittaa. Tällöin välitettäisiin vain muuttunut ja lämpötilan ylitys ja alitusraja-arvotieto ContextBrokerille. Tällöin IDAS:n MongoDB:n tulee luoda siihen tarkoitukseen laadittu tietokanta. Tähän ei kuitenkaan tällä hetkellä FIWARE-Cloud- ja NEBULA-Cloud IoT -ympäristöissä ole tarvetta palvelujen maksuttomuuden ja sensoritiedon vähäisyyden vuoksi.

Lopuksi testattiin FIWARE IoT Fog -verkon latenssia ping-komennolla Linux shell -terminaalissa (kuva 49, lohkot 2, 3, 5 ja 6).

Rakennetun FIWARE IoT Fog -verkon etuna on vain noin 10 ms:n latenssi pääkaupunkiseudulla sijaitsevalta startup-projektin laboratorion IoT-laitteiden sensoreilta Nebula-Cloudin IDAS-ohjelmistolle. Nebula-Cloudissa sijaitsee myös sensoritiedon tallennus MongoDB:hen. Jos IDAS-ohjelmisto olisi asennettu FIWARE-Cloudiin Madridiin, latenssi olisi ollut 100 ms eli kymmenkertainen. Jos on tarve 1 ms:n latenssille, se saavutetaan asentamalla IDAS-ohjelmisto vielä lähemmäksi IoT-sensoreita ja ohjattavia laitteistoja. Vertailun vuoksi 4G-teknologialla latenssi on noin 13–20 ms, ja tulevalla 5G:llä pyritään noin 1 ms:iin, joka soveltuu muun muassa älykkään liikenteen reaaliaikaiseen ohjaukseen.

6 Yhteenveto

Insinööriyössä tutkittiin, valittiin, suunniteltiin ja toteutettiin startup-yrityksen SDx-tuotekehitysympäristö pilvipalvelin-, SDN/NFV- ja IoT-teknologioin.

Työssä päästiin tavoitteisiin eli löydettiin startup-yritykselle tuotekehityspolku ja tuotekehitysalusta SDN/NFV- ja IoT-tuotteiden ja palvelujen kehittämiseksi. Itse tuotekehitysalusta on avain menestykseen, sillä palvelut tulee kyetä laajentamaan maailmanlaajuisesti ja pienin kustannuksin, jotta palvelun ylläpito olisi taloudellisesti järkevää. Palvelualustan edelleen kehittämisestä ja sen markkinoinnista eri tahoille näyttää tulevan myös yksi startup-projektin jälkeisen yrityksen päätoimialoista.

Tutkitut teknologiat ovat jo käytössä ja tulevat valloittamaan markkinoita kustannustehokkuudellaan ja ylläpidettävyydellään. Pärjätäkseen kilpailussa perinteisiä tuotteita tekevien yritysten on muututtava entistä enemmän ohjelmistopainotteisiksi yrityksiksi. Yleisesti puhutaan digitalisaatiosta, jossa digitaalisia teknologioita hyväksikäyttäen muutetaan yritysten liiketoimintamalleja, jotta voidaan luoda uusia arvonalisäys- ja ansaintamalleja. Nämä tekniikat antavat yrityksille mahdollisuuden toteuttaa uusia tuoteperheitä ja lisäominaisuuksia nykyisiin tuotteisiinsa sekä parantaa kilpailukykyä. Yleisenä huomiona voidaan todeta, että telecom-puolella ollaan siirtymässä tuotteiden valmistuksesta ohjelmistoalustojen ja palvelujen tuottamiseen. IoT-puolella tarvitaan ohjelmistojen lisäksi edelleen tuotteiden valmistusta, sillä sensorit ja yhdyskäytävät ovat fyysisiä laitteistoja ja niiden määrä lisääntyy ympäristössämme.

Työn SDN/NFV-tutkimusmateriaali perustui pitkälti teknologiajohtajamaan Yhdysvaltojen yliopistojen, kuten Princetonin ja Standfordin, julkaisuihin ja avoimiin kursseihin. Erityisesti Princetonin yliopiston Nick Feamesterin avoin Courseran ”Software Defined Networking” -kurssi antoi laajan ja syvällisen näkemyksen käsiteltävistä tekniikoista ja niiden soveltamisesta käytännössä.

Työssä yhdistettiin SDN/NFV- ja IoT-teknologiat, mikä myös eri lähteiden mukaan tuo merkittävää kilpailukykyä uusille tuotteille. Avoimen lähdekoodin ilmaiset Linux-pohjaiset työkalut tarjoavat mahdollisuuden rakentaa tuotekehitysympäristö edullisesti, laadukkaasti ja tulevaisuuden tarpeet huomioon ottaen.

Suunniteltua SDx-tuotekehitysympäristöä rakennettiin hankittuun FIWAREn pilvipalvelinympäristöön. Ensimmäinen versio tuotekehitysympäristökokonaisuudesta saatiin valmiiksi. Tuotekehitysympäristöä kehitetään jatkossa edelleen, ja se on mahdollista ottaa käyttöön esimerkiksi muissa startup-yrityksissä, sillä se tarjoaa hyvät lähtökohdat SDN/NFV- ja IoT-tuotteiden kehittämiseen. Molemmat teknologiat tarvitsevat pilvipalvelinympäristön toimiakseen. Sama pilvipalvelinympäristö sopii myös yrityksen muihin tarpeisiin. Siellä voivat sijaita esimerkiksi yrityksen kotisivut ja verkkokauppa sekä tiedostojen versiohallinta- ja tallennusjärjestelmät.

Insinööriyön jälkeen voidaan siirtyä toteuttamaan tämän insinööriyön ulkopuolella suunniteltuja IoT-tuoteperheen palveluja ja kehittämään tuotekehitysympäristöä edelleen. Palvelut ja niiden asettamat vaatimukset ohjaavat loppujen lopuksi myös tuotekehityksen käyttämiä teknologiavalintoja. Avoimet laitteistot ovat edullisia, ja avoimet ohjelmistoalustat ovat ilmaisia. Myös Euroopan unioni yhdessä Euroopan kaupunkien, yliopistojen ja yritysten kanssa rakentaa avoimia kaupunkien tietokantoja hyödyntäviä SmartCities-ohjelmistoalustoja. Tavoitteena on hyödyntää avoimia ohjelmistoja edullisten yhteiskunnallisten digitaalisten palvelujen tuottamiseksi ja Euroopan unionin globaalin kilpailukyvyn edistämiseksi.

Insinööriyö soveltuu alustavaksi tietolähteeksi koulutusmateriaalin tuottajille ja digitalisaation palvelujen kehittäjille. Hyödyntäviä tahoja voivat olla peruskoulut, ammatilliset oppilaitokset, ammattikorkeakoulut ja yliopistot sekä muut vastaavat tahot, jotka hakevat edullista, turvallista ja pitkällä aikavälillä koko Suomea ja Euroopan unionin alueen kasvua edistävää palvelualustaa, samoin tuotteita ja palveluita kehittävät startupit ja yritykset, kuten telecom- ja IoT-operaattorit. Tämä SDx-alustakokonaisuus tarjoaa laa-

jalti uusia mahdollisuuksia globaalien ja liiketaloudellisesti kannattavien palvelujen tuottamiseksi. Insinööriyön ja startup-projektin jälkeen on tarkoitus perustaa vuonna 2017 konsultti- ja tuotekehitysyritys, joka tekee oman tuoteperheen kehitys- ja myyntityön lisäksi arkkitehtuurisuunnittelua ja ohjelmistokehitysprojekteja sopimustyönä sekä tarjoaa osaamistaan edellä mainituille tahoille digitaalisten IaaS-, PaaS- ja SaaS-palvelualustojen rakentamiseksi.

Lähteet

- 1 What is Software Defined Everything – Part 1: Definition of SDx. 2016. Verkkodokumentti. SDxCentral. <<https://www.sdxcentral.com/cloud/definitions/software-defined-everything-sdx-part-1-definition>>. Luettu 26.9.2016.
- 2 Open Networking Foundation (ONF). 2014. ONF. Verkkodokumentti. <www.opennetworking.org/about/onf-overview>. Luettu 29.9.2014.
- 3 Software-defined networking (SDN). 2014. Verkkodokumentti. Wikimedia Foundation Inc. <http://en.wikipedia.org/wiki/Software-defined_networking>. Luettu 29.9.2014.
- 4 Feamster, Nick. 2015. Software-defined networking (SDN) online course. Princeton University, New Jersey USA. <<https://www.coursera/course/sdn1>>. Suoritettu 8 viikon kurssi 30.9.2015.
- 5 Feamster, Nick. 2015. Coursera SDN Virtual Machine setup. Princeton University, New Jersey USA. <<https://github.com/PrincetonUniversity/Coursera-SDN/>>. Kurssi suoritettu 30.9.2015.
- 6 Rexford, Jennifer. 2013. Cloud Trailblazers. Verkkodokumentti. Princeton University. <<https://www.youtube.com/watch?v=IG3KI9ENUFQ>>. Katsottu 22.9.2015.
- 7 Liimatainen, Esko. 2016. Pilvipalvelin, SDN/NFV- ja IoT-käyttökohde esimerkkejä. Verkkodokumentti. <https://www.dropbox.com/s/a7f47lj74p5azci/Pilvipalvelin%2C%20SDN%26NFV%20%E2%80%93%20IoT%20teknologioiden%20k%C3%A4ytt%C3%B6kohde%20esimerkkej%C3%A4_Esko%20Liimatainen_1.docx?dl=0>. Julkaistu 14.9.2016
- 8 Pilvipalvelimen määritelmä. Cloud services. 2016. Verkkodokumentti. TechTarget. <<http://searchcloudprovider.techtarget.com/definition/cloud-services>>. Luettu 14.9.2016.
- 9 ETSI SDN/NFV -standardointi. 2016. Verkkodokumentti. ETSI. <<http://www.etsi.org/technologies-clusters/technologies/nfv>>. Luettu 29.9.2014.
- 10 IETF työryhmän "Interface to the Routing Systems (i2rs)" SDN-standardi perinteisiä reititysprotokollia hyödyntäen. What are SDN Controllers (or SDN Controller Platforms)? SDxCentral. 2016. IETF. 2016. Verkkodokumentti. <<https://www.sdxcentral.com/sdn/definitions/sdn-controllers>>. Luettu 31.8.2016.
- 11 Rothenberg, Christian Esteve. 2014. Software-defined Networking and OpenFlow. Part II. Verkkodokumentti. Universität Würzburg Germany. <<http://slideplayer.com/slide/41123>> Luettu 27.10.2014.

- 12 Shenker, Scott. 2011. Software-defined networking (SDN) youtube video 1. The Future of Networking, and the Past of Protocols. Verkkodokumentti. Berkeley University California USA. Open Networking Summit. <<https://www.youtube.com/watch?v=YHeyuD89n1Y#t=26>>. Katsottu 27.10.2014.
- 13 Shenker, Scott. 2011. Software-defined networking (SDN) youtube video 2. An attempt to motivate and clarify Software-Defined Networking (SDN). Verkkodokumentti. Berkeley University California USA. Ericsson Technologys kanal. <<https://www.youtube.com/watch?v=WVs7Pc99S7w>>. Katsottu 27.10.2014.
- 14 McKeown, Nick. 2011. Software-defined networking (SDN) youtube video 3. How SDN will Shape Networking. Verkkodokumentti. Stanford University California USA. With Martin Casado, Teemu Koponen, Scott Shenker and many other. Open Networking Summit. <https://www.youtube.com/watch?v=c9-K5O_qYgA>. Katsottu 27.10.2014.
- 15 Software-defined networking (SDN) youtube video 4. 2015. ONS2015: State of SDN. Verkkodokumentti. Open Networking Summit. <https://www.youtube.com/channel/UChO2uqQqpmE_Cg5b4qiUpUg>. Katsottu 27.10.2015.
- 16 Software-defined networking (SDN) youtube video 5. 2015. ONS2015: Opening Remarks. Verkkodokumentti. Open Networking Summit. <https://www.youtube.com/channel/UChO2uqQqpmE_Cg5b4qiUpUg>. Katsottu 27.10.2015.
- 17 OpenStack. 2014. Verkkodokumentti. Wikimedia Foundation Inc. <<http://en.wikipedia.org/wiki/OpenStack>>. Luettu 29.9.2014.
- 18 OpenStack-yhteisö. 2014. Verkkodokumentti. OpenStack. <<http://www.openstack.org>>. Luettu 29.9.2014.
- 19 OpenStack tukemat hypervisorit. Feature Support Matrix. 2015. Verkkodokumentti. OpenStack. <<http://docs.openstack.org/developer/nova/support-matrix.html>>. Luettu 17.3.2015.
- 20 Raghuram, Sirish. 2015. Platform9 Company Introduction. Verkkodokumentti. Platform 9 Systems Inc. <<http://www.youtube.com/watch?v=WGO051XVlzs>>. Katsottu 25.2.2015.
- 21 Maskasky, Madhura. 2015. Platform9 OpenStack-managerin Linux (KVM) demo video. OpenStack Demo for Linux (KVM) | Platform9 Managed OpenStack. Platform 9 Systems Inc. <<http://www.youtube.com/watch?v=zy-LaXzqLv0>>. Katsottu 25.2.2015.
- 22 Maskasky, Madhura ja Raghuram, Sirish. 2015. Deployment Models for OpenStack. Verkkodokumentti. Platform 9 Systems Inc.

- <http://www.slideshare.net/OpenStack_Online/platform9-deployment-models-for-openstack>. Luettu 25.2.2015.
- 23 Nebula-hybridipilvi. 2016. Verkkodokumentti. Nebula Oy. <<http://cloud9.nebula.fi>> Luettu 26.9.2016.
 - 24 Kuikka, Riku. 2014. Nebula-hybridipilven toiminnan kuvaus. Verkkodokumentti. Nebula Oy. <<http://www.nebula.fi/fi/blog/2014/12/03/hybridipilvi-ei-pelkka-sanahirvio>> Luettu 19.3.2015.
 - 25 Port, Julie. 2013. Suurimmat pilvipalvelujen tuottajat. The 10 most Important Companies In Cloud Computing. Verkkodokumentti. Business Insider. <<http://www.businessinsider.com/10-most-important-in-cloud-computing-2013-4?op=1&IR=T>>. Luettu 09.01.2015.
 - 26 Apache CloudStack. 2016. Verkkodokumentti. Wikimedia Foundation Inc. <https://en.wikipedia.org/wiki/Apache_CloudStack>. Luettu 26.9.2016.
 - 27 Apache CloudStack yhteisön kotisivut. 2016. Verkkodokumentti. The Apache Software Foundation. <<http://cloudstack.apache.org>>. Luettu 26.9.2016.
 - 28 CloudStack ohjelmistoarkkitehtuuri. 2013. Verkkodokumentti. The Apache Software Foundation. <<https://cwiki.apache.org/confluence/display/CLOUDSTACK/Development+101>>. Luettu 29.9.2014
 - 29 OpenNebula-yhteisön kotisivut. 2015. Verkkodokumentti. OpenNebula Project. <<http://openebula.org>>. Luettu 28.5.2015.
 - 30 OpenNebula-arkkitehtuuri. 2015. Verkkodokumentti. OpenNebula Project. <http://archives.openebula.org/documentation:rel4.0_bck_20marzo:introapis>. Luettu 28.5.2015.
 - 31 Eucalyptus yhteisön kotisivut. 2015. Verkkodokumentti. Hewlett Packard Enterprise. <<http://eucalyptus.com>>. Luettu 13.3.2015.
 - 32 OpenCompute yhteisön kotisivut. 2015. Verkkodokumentti. Open Compute Project. <<http://www.opencompute.org>>. Luettu 19.2.2015.
 - 33 OpenCompute yhteisön ja Facebookin "6-pack" kytin. 2015. Verkkodokumentti. Facebook. <<https://www.youtube.com/watch?v=OyeoTPSn7b4>>. Katsottu 19.2.2015.
 - 34 OpenCompute yhteisön ja Facebookin "6-pack" kytin. 2015. Verkkodokumentti. Facebook. <<https://code.facebook.com/posts/717010588413497/introducing-6-pack-the-first-open-hardware-modular-switch>>. Luettu 19.2.2015.

- 35 Qingye, Jiang (John). 2015. Avoimien pilvipalvelinyhteisöjen vertailu. Verkkodokumentti. University of Sydney. <<http://www.qyjohn.net/?m=201504>>. Luettu 28.5.2015.
- 36 Virtualisointi. 2013. Verkkodokumentti. Wikimedia Foundation Inc. <<https://fi.wikipedia.org/wiki/Virtualisointi>> Luettu 6.9.2016.
- 37 Hypervisor. 2013. Verkkodokumentti. Wikimedia Foundation Inc. <<https://en.wikipedia.org/wiki/Hypervisor>> Luettu 6.9.2016.
- 38 XenServer virtualisointialusta. 2015. Verkkodokumentti. Citrix Systems Inc. <<http://xenserver.org>>. Luettu 28.5.2015.
- 39 XenServer arkkitehtuuri SUSE:n toteutuksella. 2015. Verkkodokumentti. SUSE LLC and contributors. <https://www.suse.com/documentation/sles11/singlehtml/book_xen/book_xen.html>. Luettu 28.5.2015.
- 40 Microsoft Hyper-V virtualisointi alusta. 2015. Verkkodokumentti. Microsoft. <<https://msdn.microsoft.com/en-us/library/cc768520%28v=bts.10%29.aspx>>. Luettu 28.5.2015.
- 41 VMware ESXi virtualisointi alusta. 2014. Verkkodokumentti. vmhub. <<http://vmhub.blogspot.fi>>. Luettu 28.5.2015.
- 42 Rantanen, Niko. 2013. Virtualisointi KVM-hypervisorilla. Verkkodokumentti. Turun ammattikorkeakoulu. <https://www.theseus.fi/bitstream/handle/10024/54246/virtualisointi_kvm-hypervisorilla.pdf?sequence=1>. Luettu 29.10.2014.
- 43 KVM-virtualisointialusta. 2011. Verkkodokumentti. IBM. <https://www.ibm.com/developerworks/community/blogs/ibmvirtualization/entry/kvm_architecture_the_key_components_of_open_virtualization_with_kvm2?lang=en>. Luettu 28.5.2015.
- 44 Linux-jakelut. 2015. Verkkodokumentti. Wikimedia Foundation Inc. <https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg>. Luettu 28.5.2015.
- 45 Vugt, Sander Van. 2014. Compare popular Linux distributions for servers. Verkkodokumentti. TechTarget. <<http://searchdatacenter.techtarget.com/feature/Compare-popular-Linux-distributions-for-servers>>. Luettu 28.5.2015.
- 46 Open vSwitch. 2015. Verkkodokumentti. Linux Foundation. <<http://openvswitch.org/features>> Luettu 09.01.2015.

- 47 Docker container. 2015. Verkkodokumentti. Docker Inc.
<<https://www.docker.com>>. Luettu 28.5.2015.
- 48 Swan, Chris. 2014. Docker container arkkitehtuurikuva. Docker drops LXC as default execution environment. Verkkodokumentti. Infoq.
>http://www.infoq.com/news/2014/03/docker_0_9>. Luettu 28.5.2015.
- 49 Mathieu, Laurent. 2014. Docker & Vagrant introductions. Verkkodokumentti. IP-PON. <<http://www.ipponusa.com/docker-vagrant>>. Luettu 28.5.2015.
- 50 SEGET, Vladan. 2015. Virtuaalikoneen ja Docker containerin vertailua. VMware Docker Integration. Verkkodokumentti. <<http://www.vladan.fr/vmware-docker-integration-and-more>>. Luettu 28.5.2015.
- 51 What is docker ? 2015. Verkkodokumentti. Docker Inc.
<<https://www.docker.com/what-docker>> Luettu 28.5.2015.
- 52 Wainewright, Phil. 2014. Virtualization is dead, long live containerization. Verkkodokumentti. Diginomica Ltd. <<http://diginomica.com/2014/07/02/virtualization-dead-long-live-containerization>>. Luettu 28.5.2015.
- 53 Docker container tuki Windowsille. Microsoft announces support for Docker container virtualization for next version of Windows Server. 2014. Verkkodokumentti. Microsoft. <<http://virtualization.info/en/news/2014/10/microsoft-announces-support-for-docker-container-virtualization-for-next-version-of-windows-server.html>>. Luettu 28.5.2015.
- 54 Johnston, Scott. 2014. Docker container tuki VMwarelle. Docker & VMware: 1+1=3.. Verkkodokumentti. Docker Inc. <<https://blog.docker.com/2014/08/docker-vmware-1-1-3>>. Luettu 28.5.2015.
- 55 Docker container tuki RedHatile. 2015. Verkkodokumentti. RedHat Inc. <<https://access.redhat.com/articles/881893>>. Luettu 28.5.2015.
- 56 Gartnerin suljettujen ja avoimien pilvipalvelujen vertailu vuodelta 2015. Magic Quadrant for Cloud Infrastructure as a Service, Worldwide. 2015. Verkkodokumentti. Gartner. <<https://www.gartner.com/doc/reprints?id=1-2EKJTD6&ct=150506&st>>. Luettu 9.8.2016.
- 57 Gartnerin suljettujen ja avoimien pilvipalvelujen vertailu vuodelta 2016. Magic Quadrant for Cloud Infrastructure as a Service, Worldwide. 2016. Verkkodokumentti. Gartner. <<https://www.gartner.com/doc/reprints?id=1-2G45TQU&ct=150519>>. Luettu 9.8.2016.
- 58 Gartnerin analyysi SDN, NFV ja White-Box kytkimien yleistymisestä. SDN hits rock bottom and FCoE is obsolete, says Gartner mages. 2015. Verkkodokumentti. Gartner.

- <http://www.theregister.co.uk/2015/07/29/sdn_enthusiasm_dives_says_gartner>. Luettu 5.8.2015.
- 59 Feamster, Nick. Rexford Jennifer. Zegura Ellen. 2013. SDN/NFV-kehityshistoria. The Road to SDN. Verkkodokumentti. Georgia Institute of Technology. Princeton University. Georgia Institute of Technology. <<https://queue.acm.org/detail.cfm?id=2560327>>. Luettu 5.8.2015.
 - 60 Duffy, Jim. 2012. SDN:n kommunikointiprotokollat. Cisco, VMware and OpenFlow fragments SDNs. Verkkodokumentti. Network World. <<http://www.networkworld.com/article/2161605/lan-wan/cisco--vmware-and-openflow-fragment-sdns.html>>. Luettu 5.8.2015.
 - 61 Wilkins, Sean. 2014. Avoimia ja kaupallisia SDN-kontrollereita. Software Defined Networking (SDN) Controllers. Verkkodokumentti. Tom's IT Pro. <<http://www.tomsitpro.com/articles/software-defined-networking-solutions,2-835-2.html>>. Luettu 5.6.2015.
 - 62 OpenFlow-kontrollerit. 2011. Verkkodokumentti. Open Networking Foundation. <<http://archive.openflow.org/wp/openflow-components>>. Luettu 25.03.2015.
 - 63 NOX/POX SDN -kontrolleriyhteisöjen kotisivut. 2015. Verkkodokumentti. NOXrepo. <<http://www.noxrepo.org/>>. Luettu 13.6.2015.
 - 64 Standfordin POX wiki sivut. 2014. Verkkodokumentti. Stanford University California USA. <<https://openflow.stanford.edu/display/ONL/POX+Wiki>>. Luettu 13.6.2015.
 - 65 Ryu SDN -kontrolleri yhteisön kotisivut. 2014. Verkkodokumentti. Ryu SDN Framework Community. <<http://osrg.github.io/ryu>>. Luettu 14.6.2015.
 - 66 Ryu SDN-kontrollerin Github sivut. 2015. What's Ryu. Verkkodokumentti. GitHub Inc. <<https://github.com/osrg/ryu>>. Luettu 14.6.2015.
 - 67 Floodlight SDN -kontrolleriyhteisön kotisivut. 2015. Floodlight Is an Open SDN Controller. Verkkodokumentti. Project Floodlight. <<http://www.projectfloodlight.org/floodlight>>. Luettu 13.6.2015.
 - 68 LoxiGen SDN -kontrolleriyhteisön kotisivut. 2015. Verkkodokumentti. Project Floodlight. <<http://www.projectfloodlight.org/indigo>>. Luettu 13.6.2015.
 - 69 OpenDaylight SDN -kontrolleriyhteisön kotisivut. OpenDaylight: Open Source SDN Platform. 2014. Verkkodokumentti. <<http://www.opendaylight.org>>. Luettu 29.9.2014.
 - 70 OpenDaylight SDN -kontrolleriyhteisön jäsenet. 2014. Verkkodokumentti. OpenDaylight. <<https://www.opendaylight.org/membership>>. Luettu 29.9.2014.

- 71 ONOS-kontrolleriyhteisön kotisivut. 2015. Verkkodokumentit. ONOS.
<<http://www.onosproject.org>>. Luettu 21.9.2015.
- 72 ONOS/SDN-IP tutkimus ja koulutus verkon laajuus vuonna 2015. Global ONOS and SDN-IP deployment. 2015. Verkkodokumentti. ONOS.
<http://onosproject.org/wp-content/uploads/2015/06/PoC_global-deploy.pdf>. Luettu 24.8.2016.
- 73 Hardesty, Linda. 2016. Infinera, Corsa & ONOS Do SDN for Europe's Research Network. Verkkodokumentti. SDxCentral.
<<https://www.sdxcentral.com/articles/news/infinera-corsa-onos-sdn-europes-research-network/2016/06>>. Luettu 24.8.2016.
- 74 ONOS Goldeneye version julkaisu materiaali. 2016. Verkkodokumentti. ONOS.
<<https://wiki.onosproject.org/display/ONOS/Release+Model>>. Luettu 24.8.2016.
- 75 Shahzad, Khurram. 2015. ONOS: The Game Changing SDN Controller? Verkkodokumentti. WordPress. <<https://kshahzadblog.wordpress.com/2015/04/11/onos-the-game-changing-sdn-controller>>. Luettu 21.9.2015.
- 76 ONOS Use Cases. 2015. Verkkodokumentti. ONOS.
<<https://wiki.onosproject.org/display/ONOS/Use+Cases>>. Luettu 21.9.2015.
- 77 Barros, Joao Marcelo. 2015. Ethernet Switching Fabric Network. Verkkodokumentti. IBM. <<http://www.slideshare.net/JoaoMarceloBarros/fabric-networking>>. Luettu 5.8.2015.
- 78 OpenFlow. 2014. Verkkodokumentti. Wikimedia Foundation Inc.
<<http://en.wikipedia.org/wiki/OpenFlow>>. Luettu 29.9.2014.
- 79 ONF OpenFlow -dokumentaatio. 2014. Verkkodokumentti. Open Networking Foundation. <<https://www.opennetworking.org/sdn-resources/openflow>>. Luettu 29.9.2014.
- 80 ONF määrittämät OpenFlow-spesifikaatiot. ONF Technical library. 2015. Verkkodokumentteja. Open Networking Foundation.
<<https://www.opennetworking.org/sdn-resources/technical-library>>. Luettu 12.06.2015.
- 81 OpenFlow Switch Specification Version 1.5.1. 2015. Verkkodokumentti. Open Networking Foundation.
<<https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf>>. Luettu 23.5.2015.
- 82 Software-Defined Networking will be a critical enabler of the Internet of Things. Jef Reed. 2015. Verkkodokumentti. Network World.
<<http://www.networkworld.com/article/2932276/sdn/software-defined-networking-will-be-a-critical-enabler-of-the-internet-of-things.html#twitter>>. Luettu 12.06.2015.

- 83 ONF:n Protocol Independent Forwarding (PIF) julkaisu. Protocol Independent Forwarding. 2015. Verkkodokumentti. Open Networking Foundation. <<https://www.opennetworking.org/protocol-independent-forwarding>>. Luettu 12.06.2015.
- 84 McKeown, Nick. 2014. Protocol Independent Forwarding (PIF). Verkkodokumentti. Open Networking Foundation. <<https://www.youtube.com/watch?v=oUv43w7ZQCM>>. Katsottu 12.06.2015.
- 85 SDN Product Directory. 2016. Verkkodokumentti. Open Networking Foundation. <<https://www.opennetworking.org/products-listing>>. Luettu 08.03.2016
- 86 SDN, NFV, and Network Virtualization Products Directory. 2015. Verkkodokumentti. SDxCentral. <<https://www.sdxcentral.com/nfv-sdn-products-directory>>. Luettu 08.03.2016.
- 87 Matsumoto, Craig. 2014. OpenFlow 2.0 Could Bring New Flexibility to Switches. Verkkodokumentti. SDxCentral. <<https://www.sdxcentral.com/articles/news/openflow-2-0-bring-new-flexibility-switches/2014/03>>. Luettu 12.06.2015.
- 88 Huaweiin Protocol Oblivious Forwarding (POF) tutkimus. Forwarding Programming in Protocol-Oblivious Instruction Set. 2014. Verkkodokumentti. Huawei. <<http://success.cse.tamu.edu/CoolSDN2014/ppt/CoolSDN14-paper4.pdf>>. Luettu 12.06.2015.
- 89 Huaweiin Protocol Oblivious Forwarding (POF) julkaisu. Huawei Shares the Source Code of Protocol-Oblivious Forwarding (POF) Project. 2013. Verkkodokumentti. Huawei. <<http://pr.huawei.com/en/news/hw-267042-pof.htm#.VxMy4vmLSM9>>. Luettu 12.06.2015.
- 90 Huaweiin Protocol Oblivious Forwarding (POF). SDN Using Huawei's unique Protocol Oblivious Forwarding POF Technology. 2014. Verkkodokumentti. Huawei. <<https://www.youtube.com/watch?v=0LQv1XHS9CU>>. Katsottu 12.06.2015.
- 91 Huaweiin Protocol Oblivious Forwarding (POF) lähdekoodit. How can POF benefit Service Providers & Network Operators. 2014. Verkkodokumentti. Huawei. <<http://www.poforwarding.org>>. Luettu 12.06.2015.
- 92 Wobker, LJ. 2015. P4-demo video. A P4 language demonstration: the "Simple Router" program in P4. Verkkodokumentti. Barefoot Networks. <<http://p4.org/wp-content/uploads/2015/04/p4-demo-video-full.mp4>>. Katsottu 12.06.2015.
- 93 P4-ohjelmoinnin lähdekoodit. P4-language. 2015. Verkkodokumentti. GitHub Inc. <<https://github.com/p4lang>>. Luettu 12.06.2015.

- 94 Shahbaz, Muhammad. 2015. OVS Datapath Specialization using P4 - OVS Conference. Verkkodokumentti. Princeton University California USA. <<https://www.youtube.com/watch?v=mqHuK1x14As>>. Katsottu 08.03.2016
- 95 Mininet verkkoemulaattorin ohjeet. Mininet Overview. 2014. Verkkodokumentti. Mininet Team. <<http://mininet.org/overview>>. Luettu 27.10.2014.
- 96 Mahler, David. 2013. Open Flow'n toiminnan havainnollistamien Mininet-verkkoemulaattorilla PC-ympäristössä. Introduction to OpenFlow. Verkkodokumentti. <<https://www.youtube.com/watch?v=l25Ukkmk6Sk>>. Katsottu 27.10.2014.
- 97 White-Box. 2014. Verkkodokumentti. Wikimedia Foundation Inc. <[http://en.wikipedia.org/wiki/White_box_\(computer_hardware\)](http://en.wikipedia.org/wiki/White_box_(computer_hardware))>. Luettu 29.9.2014.
- 98 Oracle VM VirtualBox. 2015. Verkkodokumentti. Oracle. <<https://www.virtualbox.org/wiki/Downloads>>. Ladattu 16.01.2015.
- 99 Raspberry Pi -kehityslaitteisto. 2014. Verkkodokumentti. Raspberry Pi Foundation. <<https://www.raspberrypi.org>>. Luettu 29.9.2014.
- 100 Vyatta. 2015. Verkkodokumentti. Wikimedia Foundation Inc. <<http://en.wikipedia.org/wiki/Vyatta>>. Luettu 20.1.2015.
- 101 Quagga järjestelmä arkkitehtuuri. Quagga Routing Software Suite. 2011. Verkkodokumentti. Quagga. <<http://www.nongnu.org/quagga/docs/docs-info.html#Overview>>. Luettu 20.1.2015.
- 102 Brocaden SDN/NFV -koulutus. Brocade SDN/NFV. 2015. Verkkodokumentti. Brocade Communications Systems, Inc. <<http://community.brocade.com/t5/SDN-NFV/ct-p/SdnNfv>>. Luettu 20.1.2015.
- 103 Brocade Network Function Virtualization 60 day education and certification. 2015. Verkkodokumentti. Brocade Communications Systems, Inc. <<http://www.brocade.com/products/all/network-functions-virtualization/index.page>>. Luettu 20.1.2015.
- 104 VyOS. 2015. Verkkodokumentti. Wikimedia Foundation Inc. <<http://en.wikipedia.org/wiki/VyOS>>. Luettu 20.1.2015.
- 105 VyOS project. 2015. Verkkodokumentti. VyOS project. <http://vyos.net/wiki/Main_Page>. Luettu 20.1.2015.
- 106 Network Virtualization. 2015. Verkkodokumentti. Wikimedia Foundation Inc. <https://en.wikipedia.org/wiki/Network_virtualization>. Luettu 20.1.2015.
- 107 Mäkipää Mikko. 2014. Kaupallisten SDN-kytkinten vertailu. Pilvipalveluihin liittyvät virtualisoidut lähiverkot. Verkkodokumentti. Tampereen teknillinen yli-

opisto.

<<https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/22102/Makipaa.pdf?sequence=1>>. Luettu 9.1.2015.

- 108 VMware NSX arkkitehtuuri. VMware NSX is the network virtualization platform for the Software-Defined Data Center. 2015. Verkkodokumentti. VMware Inc. <<http://www.vmware.com/products/nsx.html>>. Luettu 9.1.2015.
- 109 VMware NSX jälleenmyyjät. VMware NSX Licensing. 2015. Verkkodokumentti. VMware Inc. <<http://www.virtualizationworks.com/NSX.asp>>. Luettu 9.1.2015.
- 110 VMware Cross-Cloud arkkitehtuuri. 2016. Verkkodokumentti. VMworld 2016 US. <<http://www.vmworld.com/en/us/learning/general-sessions.html?mid=4356&eid=CVMW2000009165923>>. Luettu 2.9.2016.
- 111 Microsoft Hyper-V Extensible architecture. 2015. Verkkodokumentti. Microsoft. <<http://blog.marcosnogueira.org/extending-the-hyper-v-extensible-switch>>. Luettu 9.1.2015.
- 112 Koponen Teemu (Nicira/VMware). 2013. Stanford NetSeminar. Verkkodokumentti. Stanford University California USA. <<https://www.youtube.com/watch?v=bx0XCjJQt70>>. Luettu 22.9.2015.
- 113 Stanford NetSeminars. 2016. Verkkodokumentti. Stanford University California USA. <<http://netseminar.stanford.edu>>. Luettu 26.9.2016.
- 114 VMwaren, Ciscon ja EMC:n suomessa vuonna 2014 lanseerama, VCE (Virtual Computing Environment). 2014. Verkkodokumentti. EMC Corporation. <<http://www.vce.com>>. Luettu 29.9.2014.
- 115 Network Function Virtualization (NFV). 2015. Verkkodokumentti. Wikimedia Foundation Inc. <https://en.wikipedia.org/wiki/Network_function_virtualization>. Luettu 23.6.2015.
- 116 OPNFV. 2015. Verkkodokumentti. Open Platform for NFV Project Inc. <<https://www.opnfv.org>>. Luettu 23.6.2015.
- 117 OPNFV. Arno, Brahmaputra and Colorado versions. 2017. Verkkodokumentti. Open Platform for NFV Project Inc. <<https://www.opnfv.org/software>>. Luettu 23.6.2015.
- 118 Internet of Things (IoT). 2014. Verkkodokumentti. Wikimedia Foundation Inc. <https://en.wikipedia.org/wiki/Internet_of_things>. Luettu 29.9.2014.
- 119 Esineiden internet wiki sivut. 2014. Verkkodokumentti. Wikimedia Foundation Inc. <https://fi.wikipedia.org/wiki/Esineiden_internet>. Luettu 29.9.2014.

- 120 Intellicent Systems for a More Connected World (Intel). 2013. Verkkodokumentti. Intel. <<https://scoop.intel.com/celebrating-international-internet-of-things-day>>. Luettu 17.04.2016.
- 121 Esineiden internet mullistaa maailmaa yhtäpaljon kuin sähkön ja veden tulo. 2016. Verkkodokumentti. Yle. <http://yle.fi/uutiset/esineiden_internet_mullistaa_maailmaa_yhta_paljon_kuin_sahkon_ja_tietokoneiden_tulo_ensin_pitaa_ratkaista_tyhjenevien_akkujen_ongelma/8302027>. Luettu 16.9.2016.
- 122 Gartnerin analyysi Internet of Things (IoT):n yleistymisestä. Gartner's 2015 Hype Cycle for Emerging Technologies. Internet of Things (IoT). 2015. Verkkodokumentti. Gartner. <<http://www.gartner.com/newsroom/id/3114217>>. Luettu 2.12.2015.
- 123 Gartnerin analyysi Software Defined Anything (SDx):n yleistymisestä. Gartner's 2016 Hype Cycle for Emerging Technologies. Software Defined Anything (SDx). 2016. Verkkodokumentti. Gartner. <<http://www.forbes.com/sites/louiscolombus/2016/08/21/gartner-hype-cycle-for-emerging-technologies-2016-adds-blockchain-machine-learning-for-first-time/#750fc4461ef2>>. Luettu 26.9.2016.
- 124 Gartner, IoT Standards and Ecosystems. Gartner Identifies the top 10 Internet of Things Technologies for 2017 and 2018. Title: IoT Standards and Ecosystems. 2016. Verkkodokumentti. Gartner. <<http://www.gartner.com/newsroom/id/3221818>>. Luettu 22.9.2016.
- 125 Maailman väkiluku. 2016. Verkkodokumentti. Wikimedia Foundation Inc. <https://fi.wikipedia.org/wiki/Maailman_v%C3%A4kiluku>. Luettu 26.9.2016.
- 126 Nokian 4G LTE VoLTE NFV julkaisu vuodelta 2014. Nokian Networks releases commercial NFV solution for VoLTE. 2014. Verkkodokumentti. Nokia Networks. <<http://www.telecomtv.com/articles/nfv/nokia-networks-releases-commercial-nfv-solution-for-volte-11704>>. Luettu 29.9.2014.
- 127 Nokian AirScale 5G tukiasema julkaisu vuodelta 2016. 2016. Verkkodokumentti. Nokia. <<http://www.digitoday.fi/tiede-ja-teknologia/2016/04/08/5gn-matkassa-on-monta-mutkaa--nain-toimii-nokia/20163781/66>>. Luettu 10.4.2016.
- 128 Suomen DIGILEn Internet of Things (IoT) tutkimus. 2014. Verkkodokumentti. DIGILE. <<http://www.internetofthings.fi>>. Luettu 29.9.2014.
- 129 Finnish Industrial Internet Forum - FIIF. 2016. Verkkodokumentti. FIIF. <<http://www.fiif.fi/main.html>>. Luettu 17.04.2016.
- 130 Nokia Growth Partners (NGP) Announces \$350M Fund Focused On IoT. Nasha Lomas. 2016. Verkkodokumentti. TechCrunch.

- <<https://techcrunch.com/2016/02/21/nokia-growth-partners-announces-350m-fund-focused-on-iot>>. Luettu 26.09.2016.
- 131 Gartner Says IoT Adoption Is Driving the Use of Platform as a Service. 2016. Verkkodokumentti. Gartner. <<http://www.gartner.com/newsroom/id/3241817>>. Luettu 26.09.2016.
- 132 Turck Matt. IoT-maisemakuva. Internet of Things Landscape. 2016. Verkkodokumentti. FirstMark Capital. <<http://mattturck.com/wp-content/uploads/2016/03/Internet-of-Things-2016.png>>. Luettu 7.9.2016.
- 133 Salman Tara. 2015. Internet of Things Protocols and Standards. Verkkodokumentti. Washington University in St.Louis. <http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot>. Luettu 07.09.2016.
- 134 WLAN-standard. 2016. Verkkodokumentti. Wikimedia Foundation Inc. <<https://fi.wikipedia.org/wiki/WLAN>>. Luettu 26.09.2016.
- 135 ZigBee-standard. 2016. Verkkodokumentti. Wikimedia Foundation Inc. <<http://fi.wikipedia.org/wiki/ZigBee>>. Luettu 26.9.2016.
- 136 Kallioniemi Tapio. 2009. ZigBee standardin toiminta ja periaatteet. Verkkodokumentti. Tampereen ammattikorkeakoulu. <https://www.theseus.fi/bitstream/handle/10024/13729/Kallioniemi_Tapio.pdf?sequence=1>. Luettu 26.9.2016.
- 137 Bluetooth-versions. 2016. Verkkodokumentti. Wikimedia Foundation Inc. <<https://en.wikipedia.org/wiki/Bluetooth>>. Luettu 26.9.2016.
- 138 Bluetooth 4.0. 2016. Verkkodokumentti. Wikimedia Foundation Inc. <http://en.wikipedia.org/wiki/Bluetooth_low_energy>. Luettu 26.9.2016.
- 139 What is Bluetooth 5? The new wireless technology standard explained. 2016. Verkkodokumentti. Trusted Reviews. <<http://www.trustedreviews.com/opinions/what-is-bluetooth-5>>. Luettu 26.9.2016.
- 140 LoRaWLAN-standard. 2016. Verkkodokumentti. The Things Networking Wiki. <<https://www.thethingsnetwork.org/wiki/LoRaWAN/Overview>>. Luettu 7.9.2016.
- 141 LoRaWLAN alliance. 2016. Verkkodokumentti. LoRa Alliance. <<https://www.lora-alliance.org>>. Luettu 7.9.2016.
- 142 WirePas Connectivity-sensoriverkko. 2016. Verkkodokumentti. Wirepas Oy. <<https://www.wirepas.com>>. Luettu 7.9.2016.

- 143 SIGFOX-sensoriverkko. 2016. Verkkodokumentti. Sigfox.
<<https://www.sigfox.com>>. Luettu 7.9.2016.
- 144 5G-verkko mahdollistaa tieteiselokuvista tutun maailman. 2016. Verkkodokumentti. Kauppalehti. <<http://studio.kauppalehti.fi/sonera-b2d/5g-verkko-mahdollistaa-tieteiselokuvista-tutun-maailman-laakari-voi-leikata-potilaan-etana-tai-virtuaalilaseilla-voi-kokea-new-yorkin-koto-suomesta>>. Luettu 30.9.2016.
- 145 Teollinen internet 4.0 (RAMI 4.0). An Architectural Model for Industry 4.0. Object Management Group (OMG). 2015. Verkkodokumentti.
<<http://www.omg.org/news/meetings/tc/berlin-15/special-events/mfg-presentations/adolphs.pdf>>. Luettu 10.1.2017.
- 146 FITMAN IoT-arkkitehtuuri. 2015. Verkkodokumentti. Euroopan unioni.
<http://ec.europa.eu/information_society/newsroom/image/document/2015-48/workshop_report_platforms_oct15_finaldocx_12361.pdf>. Luettu 10.1.2017.
- 147 Potier Benoit. 2016. Digital Single Market (DSM), A New Path To European Leadership. Verkkodokumentti. LinkedIn.
<https://www.linkedin.com/pulse/digital-single-market-new-path-european-leadership-beno%C3%A9t-potier?trk=v-feed&lipi=urn%3Ali%3Apage%3Ad_flagship3_feed%3BnxRIQZZERS%2BhWskzqHK0gA%3D%3D>. Luettu 30.9.2016.
- 148 Eurooppaan yhtenäiset digitaaliset markkina. Putting the Digital Single Market at the heart of Europe. DIGITAL SINGLE MARKET. 2016. Verkkodokumentti. European Commission. <<https://ec.europa.eu/digital-single-market/en>>. Luettu 23.9.2016.
- 149 AIOTI. The Alliance for Internet of Things Innovation. DIGITAL SINGLE MARKET. 2016. Verkkodokumentti. European Commission.
<<https://ec.europa.eu/digital-single-market/en/alliance-internet-things-innovation-aioti>>. Luettu 23.9.2016.
- 150 Euroopan alueen standardisointi järjestöjä. European Committee for Standardization (CES) ja European Committee for Electrotechnical Standardization (CENELEC). 2017. Verkkodokumentti. CEN CENELEC.
<<http://www.cencenelec.eu/aboutus/Pages/default.aspx>>. Luettu 16.1.2017.
- 151 ETSI standardointi. Teknologia klusterit. ETSI Technology Clusters. 2016. Verkkodokumentti. ETSI. <<http://www.etsi.org/technologies-clusters>>. Luettu 29.9.2014.
- 152 ETSI IoT-standardointi. ETSI Internet of Things. Standardizing the IoT. 2016. Verkkodokumentti. ETSI. <<http://www.etsi.org/technologies-clusters/technologies/internet-of-things>> Luettu 23.09.2016

- 153 Arkko Jari. 2016. IETF IoT-standardointi. An Interoperable Internet Of Things. Verkkodokumentti. IETF. <<https://www.ietf.org/blog/2016/01/an-interoperable-internet-of-things>> Luettu 23.09.2016
- 154 Euroopan unionin FIWARE yhteisö. FIWARE. 2016. Verkkodokumentti. Fiware Foundation. <<https://www.fiware.org>> Pääsivut. Luettu 28.5.2015.
- 155 FIWAREn visio. Overall FIWARE vision. 2016. Verkkodokumentti. Fiware Foundation. <http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Overall_FIWARE_Vision#Summary_of_key_concepts_introduced> Luettu 02.06.2016
- 156 FIWARE OFNIC-controller. 2014. Verkkodokumentti. Fiware Foundation. <https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.OpenSpecification.I2ND.NetIC_R3>. Luettu 02.06.2016.
- 157 FIWARE SmartCities alusta. FIWARE: Value proposition for Smart Cities. 2014. Verkkodokumentti. Fiware Foundation. <<http://www.slideshare.net/JuanjoHierro/fiware-a-standard-platform-for-smart-cities>> Luettu 23.09.2016.
- 158 FIWARE IoT-pino. 2014. Verkkodokumentti. Fiware Foundation. <<https://fiware-iot-stack.readthedocs.io/en/latest/walkthrough/index.html>> Luettu 08.09.2016.
- 159 FIWARE SmartCity, SmartSantander toteutus. 2016. Verkkodokumentti. Imagery. <<http://maps.smartsantander.eu>> Luettu 08.09.2016.
- 160 FIWARE SmartEnergy, Fortum Espoon kaukolämpöverkko toteutus. 2016. Verkkodokumentti. Cyberlightning Ltd. <<https://cyberlightning.com/demos>> Luettu 08.09.2016.
- 161 Microservices. 2016. Verkkodokumentti. Wikimedia Foundation Inc. <<https://en.wikipedia.org/wiki/Microservices>>. Luettu 24.8.2016.
- 162 Chakrabarti Sudip. 2016. Micro-Services arkkitehtuuri tulee yleistymään. In the land of microservices, the network is the king (maker). Verkkodokumentti. Lightspeed Venture Partners. <<https://medium.com/lightspeed-venture-partners/in-the-land-of-microservices-the-network-is-the-king-maker-37de7ec4119a#yfp5xtt7n>>. Luettu 24.8.2016.
- 163 Nebulan Microservices PaaS-palvelu. 2016. Verkkodokumentti. Nebula. <<http://cloud9.nebula.fi/app.html>>. Luettu 26.9.2016.
- 164 RedHatin Microservices PaaS-palvelu. RedHat OPENSIFT Container Platform. 2016. Verkkodokumentti. RedHat. <<https://www.openshift.com>>. Luettu 26.9.2016.

- 165 What is Kubernetes? 2016. Verkkodokumentti. Kubernetes.
<<http://kubernetes.io/docs/whatisk8s>>. Luettu 26.9.2016.
- 166 Kontenan Microservices PaaS-palveluohjelmisto. The Developer Friendly
Container & Microservices Platform. 2016. Verkkodokumentti. Kontena.
<<https://www.kontena.io>>. Luettu 26.9.2016.
- 167 Microservices PaaS-palveluohjelmisto Kontena lyhentää microservices
PaaS ympäristön rakentamiseen kuluvaan aikaa. Suomalaisfirma keksi lääkkeen
Docker tuskaan ja keräsi 1,75 miljoonaa. 2016. Verkkodokumentti. Kontena.
<http://www.tivi.fi/Kaikki_uutiset/suomalaisfirma-keksi-laakkeen-docker-tuskaan-ja-kerasi-1-75-miljoonaa-juristimme-ihmettelivat-6583416>. Luettu 26.9.2016.
- 168 Digitalisaatio. 2016. Verkkodokumentti. Wikimedia Foundation Inc.
<<https://fi.wikipedia.org/wiki/Digitalisaatio>>. Luettu 26.9.2016.
- 169 Freeboard:n graafinen käyttöliittymä IoT-sensoritiedon havainnollistami-
seksi. 2016. Verkkodokumentti. Bug Labs, Inc. <freeboard.io>. Luettu 26.9.2016.
- 170 Ubuntu-MATE-käyttöjärjestelmä. 2016. Verkkodokumentti. Canonical Ltd.
<<https://ubuntu-mate.org>>. Luettu 26.9.2016.
- 171 Raspbian-käyttöjärjestelmä. 2016. Verkkodokumentti. Raspberry Pi
Foundation. <<https://www.raspberrypi.org>>. Luettu 26.9.2016.
- 172 Windows 10 IoT -käyttöjärjestelmän- ja Adafruit IoT-sensorilaiterympäristö.
2016. Verkkodokumentti. Microsoft. <<https://developer.microsoft.com/fi-fi/windows/iot>>. Luettu 26.9.2016.
- 173 Arduino-käyttöjärjestelmä. 2016. Verkkodokumentti. Arduino.
<<https://www.arduino.cc>>. Luettu 26.9.2016.
- 174 CentOS 7-käyttöjärjestelmä. 2016. Verkkodokumentti. RedHat.
<<https://www.centos.org/>>. Luettu 26.9.2016.
- 175 SDN ONOS -ohjelmisto. 2016. Verkkodokumentti. ON.Lab.
<<http://onosproject.org>>. Luettu 26.9.2016.
- 176 SDN Open vSwitch -ohjelmisto. 2016. Verkkodokumentti. A Linux Foun-
dation Collaborative Project. <<http://openvswitch.org>>. Luettu 26.9.2016.
- 177 Adafruit IoT -ohjelmisto. 2016. Verkkodokumentti. Adafruit.
<<https://www.adafruit.com>>. Luettu 26.9.2016.